UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

Technical Report #26

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*

Multi-Item (s,S) Inventory Systems with a Service Objective

5. TYPE OF REPORT & PERIOD COVERED

Technical

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

J. Christopher Mitchell

8. CONTRACT OR GRANT NUMBER(s)

N00014-78-C-0467

9. PERFORMING ORGANIZATION NAME AND ADDRESS

University of North Carolina at Chapel Hill
Chapel Hill, N.C.

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Mathematical and Information Sciences Division
Office of Naval Research, Code 434
Arlington, VA 22217

12. REPORT DATE

November 1983

13. NUMBER OF PAGES

157 and 43 (Appendix)

14. MONITORING AGENCY NAME & ADDRESS*(If different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

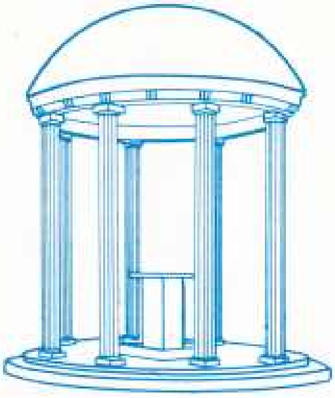
20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report considers a Multi-item (s,S) Inventory System in which shortage costs are replaced by a stockout probability constraint. Necessary and sufficient conditions for a policy to be optimal are derived, and a computational efficient algorithm is developed. Computation experience indicates the operating costs can be reduced significantly.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

UNIVERSITY OF NORTH CAROLINA university.
AT CHAPEL HILL



SCHOOL OF BUSINESS ADMINISTRATION
AND
CURRICULUM IN
OPERATIONS RESEARCH AND SYSTEMS ANALYSIS

MULTI-ITEM (s,S) INVENTORY SYSTEMS
WITH A SERVICE OBJECTIVE

Technical Report #26

J. Christopher Mitchell

November 1983

Work Sponsored By

Office of Naval Research (N00014-78-C-0467)

Decision Control Models in Operations Research

Harvey M. Wagner
Principal Investigator
School of Business Administration
University of North Carolina at Chapel Hill

FOREWARD

As part of the ongoing program in "Decision Control Models in Operations Research," Mr. J. Christopher Mitchell has studied an inventory model that uses a system-wide stockout constraint rather than individual shortage costs. He shows how to easily compute policies. The computational experience demonstrates that inventory investment can be reduced more than 20% using this approach. The report also suggests how to implement the algorithm in a real-life system that may contain thousands of stocked items. Other related reports dealing with the program are given on the following pages.

MacCormick, A. (1974), Statistical Problems in Inventory Control, ONR and ARO Technical Report 2, December 1974, School of Organization and Management, Yale University, 244 pp.

Estey, A. S. and R. L. Kaufman (1975), Multi-Item Inventory System Policies Using Statistical Estimates: Negative Binomial Demands (Variance/Mean = 9), ONR and ARO Technical Report 3, September 1975, School of Organization and Management, Yale University, 85 pp.

Ehrhardt, R. (1975), Variance Reduction Techniques for an Inventory Simulation, ONR and ARO Technical Report 4, September 1975, School of Organization and Management, Yale University, 24 pp.

Kaufman, R. (1976), Computer Programs for (s,S) Policies Under Independent or Filtered Demands, ONR and ARO Technical Report 5, School of Organization and Management, Yale University, 65 pp.

Kaufman, R. and J. Klincewicz (1976), Multi-Item Inventory System Policies Using Statistical Estimates: Sporadic Demands (Variance/Mean = 9), ONR and ARO Technical Report 6, School of Organization and Management, Yale University, 58 pp.

Ehrhardt, R. (1976), The Power Approximation: Inventory Policies Based on Limited Demand Information, ONR and ARO Technical Report 7, June 1976, School of Organization and Management, Yale University, 106 pp.

Klincewicz, J. G. (1976), Biased Variance Estimators for Statistical Inventory Policies, ONR and ARO Technical Report 8, August 1976, School of Organization and Management, Yale University, 24 pp.

Klincewicz, J. G. (1976), Inventory Control Using Statistical Estimates: The Power Approximation and Sporadic Demands (Variance/Mean = 9), ONR and ARO Technical Report 9, November 1976, School of Organization and Management, Yale University, 52 pp.

Klincewicz, J. R. (1976), The Power Approximation: Control of Multi-Item Inventory Systems with Constant Standard-Deviation-To-Mean Ratio for Demand, ONR and ARO Technical Report 10, November 1976, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 47 pp.

Kaufman, R. L. (1977), (s,S) Inventory Policies in a Nonstationary Demand Environment, ONR and ARO Technical Report 11, April 1977, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 155 pp.

Ehrhardt, R. (1977), Operating Characteristic Approximations for the Analysis of (s,S) Inventory Systems, ONR and ARO Technical Report 12, April 1977, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 109 pp.

Schultz, C. R., R. Ehrhardt, and A. MacCormick (1977), Forecasting Operating Characteristics of (s,S) Inventory Systems, ONR and ARO Technical Report 13, December 1977, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 47 pp.

Schultz, C. R. (1979), (s,S) Inventory Policies for a Wholesale Warehouse Inventory System, ONR Technical Report 14, April 1979, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 75 pp.

Schultz, C. R. (1980), Wholesale Warehouse Inventory Control with Statistical Demand Information, ONR Technical Report 15, December 1980, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 74 pp.

Ehrhardt, R. and G. Kastner (1980), An Empirical Comparison of Two Approximately Optimal (s,S) Inventory Policies, Technical Report 16, December 1980, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 22 pp.

Ehrhardt, R. (1980), (s,S) Policies for a Dynamic Inventory Model with Stochastic Lead Times, Technical Report 17, December 1980, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 20 pp.

Mosier, C. (1981), Revised (s,S) Power Approximation, Technical Report 18, February 1981, School of Business Administration, University of North Carolina at Chapel Hill, 18 pp.

Blazer, D. and M. McClelland (1981), An Inventory Model for Special Handling of Extreme Value Demands, Technical Report 19, December 1981, School of Business Administration, University of North Carolina at Chapel Hill, 10 pp.

Mitchell, J.C. (1982), Choosing Single-Item Service Objectives in a Multi-Item Base-Stock Inventory System, Technical Report 20, School of Business Administration, University of North Carolina at Chapel Hill, 30 pp.

Blazer, D. (1983), *Operating Characteristics for an Inventory Model That Special Handles Extreme Value Demand*, Technical Report #21, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 15pp.

Blazer, D. (1983), *Evaluation of a "Large Pop" Filtering Rule for Inventory Management Systems*, Technical Report #22, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 32pp.

Blazer, D. (1983), *Testing the Cost Effectiveness of an Inventory Filtering Rule Using Empirical Data*, Technical Report #23, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 24pp.

Mazzola, J.B., W.F. McCoy and H.M. Wagner (1983), Algorithms and Heuristics for Variable-Yield Lot-Sizing, ONR Technical Report 24, March 1983, School of Business Administration, University of North Carolina at Chapel Hill, 18 pp.

Blazer, D. (1983), Implementation Strategy for an Inventory Filtering Rule, ONR Technical Report 25, School of Business Administration University of North Carolina at Chapel Hill, 20 pp.

ABSTRACT

J. CHRISTOPHER MITCHELL. Multi-Item (s,S) Inventory Systems with a Service Objective (Under the direction of HARVEY WAGNER.)

This paper considers a multi-item (s,S) inventory system. The model differs from standard treatments in that shortage costs are replaced by stockout probability constraints to be satisfied in every period. The value of such a model is that it is often easier to express service objectives in terms of stockout probability constraints than it is to specify shortage costs. Specifically, system service is defined in terms of a weighted average of single-item stockout probabilities. An optimal policy minimizes system cost while satisfying a constraint on system service. Using standard single-item approximations, necessary and sufficient conditions for a policy to be optimal (for the approximate model) are derived, and a computationally efficient algorithm, the Generalized Knapsack Duality (GKD) Algorithm, is developed to find such a policy. Computational experience on inventory systems typical of many found in the real world indicates that operating costs can be reduced significantly when this model is used rather than the simpler uniform service model often used by managers.

Sensitivity experience on inventory systems with a structure typical of many real-world inventory systems is reported.

Specifically, the sensitivity of the GKD Algorithm to changes in the reorder quantities and lower bounds on feasible policies is reported. This experience suggests that, as in the single-item case, very accurate specifications of the reorder quantities are unnecessary. Recommendations are made to specify the lower bounds as high as possible while achieving significant cost savings below that of the Identical Service Approach. This recommendation is consistent with the objectives of many managers, and in certain cases improves algorithm performance.

This paper also reports computational experience with sampling schemes of various sizes for large-scale inventory aggregation. This experience indicates that inventory systems typical of many found in the real world can be well-managed based on decisions made from a relatively small number of items from the systems. In particular, this experience suggests that a central uniform sample of about 32 items is sufficient to make accurate decisions for the entire inventory system. The paper concludes with a detailed implementation procedure using the GKD Algorithm with sampling to manage a large-scale inventory system.

CONTENTS

I. INTRODUCTION 1

1. Inventory Theory with Shortage Costs 3
2. Inventory Theory with Service-Level Constraints 10
3. Multi-Item Inventory System Aggregation 15

II. THE GENERALIZED KNAPSACK DUALITY (GKD) ALGORITHM 18

1. Introduction 18
2. Analysis 21
3. GKD Algorithm 40
4. Policy Performance 46
5. GKD Algorithm Modification when D_i is Small 50
6. Conclusions 52
   - Notation 53
   - Frequently Used Relations 55
   - Theorems and Lemmas 55
   - Appendix 56

III. COMPUTATIONAL EXPERIENCE WITH THE GKD ALGORITHM 57

1. Introduction 57
2. Total Cost Savings 57
3. Single-Item Service-Levels 62
4. Items with D_i Too Small 63
5. Conclusions 65

IV. SENSITIVITY EXPERIENCE WITH THE GKD ALGORITHM 80

1. Introduction 80
2. Sensitivity Experience for D_i 81
3. Sensitivity Experience for Lower Bounds on S_i 85
4. Sensitivity Experience for Sampling Schemes for Inventory Aggregation 92
5. Sensitivity Experience for the Sample Size for Inventory Aggregation 99
6. Conclusions 104
   - Appendix 134

V. CONCLUSION: IMPLEMENTATION OF THE GKD ALGORITHM WITH SAMPLING FOR A LARGE-SCALE INVENTORY SYSTEM 146

1. Introduction 146

2. Implementation Procedure 148

3. Example 151

4. Conclusions, and Future Research Directions 152

- Appendix 158

REFERENCES 195

ACKNOWLEDGEMENTS

I want to thank above all my dissertation advisor, Professor Harvey Wagner. His extensive experience and sharp insights have been invaluable in determining both the content and direction of this research. To Professor Wagner I also offer special thanks for always helping me to see "the forest for the trees."

I want to thank Professor Richard Ehrhardt for his generous time and interest in my work, as well as his technical expertise in inventory theory. Many of our conversations helped shape the theoretical development of this work.

I want to thank Professor Jon Tolle for the considerable and valuable time he invested in my work. His expertise in nonlinear programming as well as his careful reading of earlier manuscripts resulted in significant improvements in this work. The mathematical clarity of Chapter II was greatly enhanced by his suggestions.

I want to thank my typist, Miss Kristina Hanan. She has always done a careful and excellent job, even in the most tedious parts of the paper.

Special thanks are given to the Office of Naval Research, who partially supported this research under Grant #N00014-78-C-0467.

I. INTRODUCTION

This paper examines the problem of specifying single-item service objectives in a multi-item inventory system subject to an overall (or system-wide) service-level constraint. We develop a computationally efficient algorithm for doing so when the number of items in the inventory system is of moderate size. We also investigate methods of aggregating very large inventory system so that they are computationally more manageable. For both of these problems we show by extensive numerical investigations that our methods can result in a substantial total cost savings over methods which specify uniform service-level objectives for all items in the inventory system.

This chapter is a non-technical survey of the literature on both exact and approximate methods for inventory management. Section 1 describes the theoretical and computational difficulties involved in single-item inventory management as well as methods that have been used to deal with these difficulties. Certainly a multi-item inventory system will inherit these difficulties, and so we use such single-item methods in our model when appropriate.

Section 2 discusses service-level constraints in single-item inventory systems in the same spirit as Section 1. We also survey the methods that have been used in multi-item inventory systems with service-level constraints (or similar models) in order to motivate the direction of our research.

Section 3 surveys the literature on the problem of large-scale inventory aggregation. In particular, we discuss an empirically-observed structure that is typical for many real-world inventory systems.

Chapter II contains the mathematical derivation of our algorithm to specify single-item service objectives in multi-item inventory systems with a system-wide service-level constraint. There are certain properties the items in the inventory system must satisfy in order to guarantee algorithm convergence. We illustrate algorithm performance on some two-item inventory systems which contain only items satisfying these properties. We also discuss our recommendations when there are items in the inventory system that do not satisfy all these properties.

Chapter III contains a numerical investigation of some 32-item inventory systems (with some items that do not satisfy the above-mentioned properties) which reflect a structure often observed in practice. We show that there is a significant cost decrease when using our algorithm to specify operating policies rather than using the popular method of specifying operating policies which give uniform service.

Chapter IV contains a numerical sensitivity investigation of 32-item and 128-item inventory systems (with the same structure mentioned), and a numerical investigation of sampling schemes for inventory aggregation.

Chapter V concludes this paper with a detailed implementation procedure of the GKD Algorithm with sampling to manage a large-scale inventory system. We illustrate this procedure with a 512-item inventory system, and offer conclusions and directions for future research.

1. INVENTORY THEORY WITH SHORTAGE COSTS

1.1 Model Formulation and Optimal Policies

We consider the periodic-review dynamic inventory model. We give a non-technical survey of the literature on this model, leaving relevant mathematical descriptions for the next chapter.

At the beginning of each period n, n=1,2,..., the inventory position (stock on hand plus stock on order) is reviewed, at which time a positive order may be placed. An order placed in period n is received and paid for in period n+k, where k, the leadtime, is a fixed positive integer. There is charged a fixed ordering cost K plus a linear ordering cost c. After the inventory position is reviewed there is a random demand ξ_n, where $\xi_1, \xi_2 ....$ are independent and identically distributed with cumulative distribution function (cdf) Φ, density ϕ, mean μ, and finite standard deviation σ. After demand realization, a linear holding cost h is assessed for each unit of inventory on hand. If demand exceeds the inventory on hand, the excess demand is completely backlogged, and a linear shortage cost p is assessed for each unit of backlogged demand. Then period n+1 is entered, repeating the process. Future costs are discounted at the single-period rate β with $0 \leq \beta \leq 1$ (β=0 corresponds to a single-period model and β=1 corresponds to an undiscounted model), and the objective is to minimize the total expected cost of operating the system over a prescribed horizon.

This model can be formulated as a dynamic program. Doing so and using induction, Scarf [1960] showed that under the assumptions of finite horizon length T and convex differentiable expected single-period holding costs, there is an optimal policy of the form (s_n,S_n), n=1,...,T. This policy requires that if the inventory position at the beginning of period n is less than s_n, an order is placed to raise it to S_n. Otherwise, no order is placed. Zabel [1962] extended this result to the case when the single-period holding and shortage costs are not differentiable. Earlier Karlin [1958a] showed the optimality of (s_n,S_n) policies under much stronger assumptions than Scarf; and Veinott [1966a] and Schäl [1976] extended Scarf's result to models with more general cost functions. If, however, holding and shortage costs are linear, as in our model, Scarf's result is sufficient to guarantee the optimality of (s_n,S_n) policies.

A basic difficulty in implementing these policies is that they usually vary from period to period and hence require tremendous computational effort. The situation becomes much simpler in the infinite horizon case. In this case, to minimize total expected cost we must impose $\beta<1$ or else for almost any policy this cost is infinite. One can consider an undiscounted model ($\beta=1$) if the objective criterion is changed to the average expected cost per period. If this is the criterion, Iglehart [1963a] and [1963b] proved that given the cost structure we have assumed, in both the discounted and undiscounted case, respectively, there is an optimal policy that is stationary (s,S). Such a policy requires that when the inventory position in any period falls below s, an order is placed to bring it up to S. Thus, there are only two numbers to be computed. One must, of course, assume that the true horizon length is of sufficient length to be reasonably approximated by an infinite horizon model.

The computational procedures to find exact stationary (s,S) policies generally involve deriving a closed form expression for the expected infinite horizon average cost per period given that the policy (s,S) is being followed in every period, and then minimizing this cost over all (s,S) policies. There have been several methods used to derive this steady-state cost. Karlin [1958a] derived it using linear operator theory and in [1958b] using renewal theory, which was simplified by Sahin [1982]. We mention that in both references cited, Karlin also derived a closed form expression for the optimal policy when the demand distribution is exponential. Morse [1959] derived the expected average cost per period using Markov processes theory, and Leneman and Beutler [1969] used a stationary point process approach. The first general and exact computational procedure to find an optimal stationary (s,S) policy is found in Veinott and Wagner [1965]. Assuming that the demand distribution is discrete, they derived the steady-state cost and then minimized it using finite difference calculus. This algorithm has been programmed in PL-1, and the documentation can be found in Kaufman [1976]. Bell [1970] has suggested an improvement to the Veinott-Wagner algorithm using optimal stopping rule theory, and recently Federgruen and Zipkin [1981] have described a completely different approach using a policy iteration technique for Markov decision processes. A method similar to this was suggested earlier by Johnson [1968]. In a somewhat different spirit, Sivazlian [1971] uses Gaussian quadrature methods to create graphs to compute optimal (s,S) policies.

There are three significant difficulties involved in finding optimal (s,S) policies. We discuss these in order to introduce the various approximations to optimal (s,S) policies that have been suggested in the literature.

First, the steady-state costs always involve the renewal function of the demand process. If the demand distribution is discrete, Veinott and Wagner [1965] showed that the renewal function can be evaluated recursively, although this can be computationally expensive when S is very large. If the demand distribution is continuous, there is no general computationally efficient method of even approximating the renewal function (the exception being the exponential distribution).

Second, although the steady-state cost is, in general, convex in S, it is not even unimodal in D=S-s. Thus, local minima may not be global minima. Veinott and Wagner [1965] dealt with this by establishing bounds on the optimal (s,S) policy and examining all values of D among these policies to find a global minimum. Recently Sahin [1982] has shown that for a class of distributions the steady-state cost is pseudo-convex in D and S, so local minima are global minima. It is not clear, however, that this class contains distributions useful for inventory models.

Third, exact algorithms require that the demand distribution be fully known. Many times in practice only a couple of moments will be known, and even these values may be statistical estimates.

1.2 Approximations to Optimal Policies

In light of the difficulties involved in finding optimal policies, numerous approximation procedures have been suggested. We describe the best of these.

For an undiscounted infinite horizon inventory model the ordering cost c does not appear in the cost functions. Essentially this is because all demand must eventually be satisfied, and the cost of doing so is undiscounted. Thus, there are three components of total expected cost, one associated with each of the parameters K, h and p. They can be denoted, respectively, as the expected replenishment cost, expected holding cost and expected shortage cost. Another quantity of interest is the service-level, the frequency of periods without any backorder (roughly, the steady-state probability of meeting all demand in any given period).

One of the earliest and best approximations is the Normal Approximation of Wagner [1975, pp.831-836]. It is based on asymptotic renewal theory results of Roberts [1962], the empirically-based heuristic when D=S-s is small discussed in Wagner et al [1965], and the assumption that demand is well-enough approximated by a Normal distribution. There were extensive numerical investigations of this approximation done in MacCormick [1974], Estey and Kaufman [1975], MacCormick et al [1977], and MacCormick [1977] in which the actual underlying demand distribution was assumed to be Negative Binomial or Poisson. The Normal Approximation in general performed very well even when statistical estimates of the demand distribution mean and variance were used (although degradation was greater in this case). The greatest degradation in performance was observed when the coefficient of variation $\frac{\sigma}{\mu}$ was large (in which case there is a non-negligible probability of negative demand for a Normal random variable), and when both $\frac{\sigma^2}{\mu}$ and $\frac{p}{h}$ were large (the latter corresponds to a high service level). When $\frac{\sigma^2}{\mu}$ and $\frac{p}{h}$ were large, the expected shortage cost and service-level degraded much more than the expected replenishment and holding costs.

Using the general approach of Norman and White [1968] of computing approximately optimal policies in Markov Decision processes by replacing the probability distributions by their moments, Porteus [1979] developed an algorithm to compute approximately optimal (s,S) policies. However, the empirical examination done in Freeland and Porteus [1980] showed that it was not much better than the Normal Approximation, and was a lot more work to compute.

Using regression models suggested by both asymptotic analysis and empirical observations, Ehrhardt [1976] developed the Power Approximation. Extensive numerical investigations were performed by Ehrhardt [1976] (compare Ehrhardt [1978]) and Klincewicz [1976a] and [1976b]. These investigations showed that not only are the approximations especially easy to compute, require only the first two moments of the demand distribution and very useful for sensitivity analysis, but they are extremely accurate over a wide range of parameter settings. The Power Method's performance was always superior to the performance of the Normal Approximation. The only cases when the Power Approximation did not perform well occurred in the expected shortage cost and service-level when statistical demand moments were used and $\frac{p}{h}$ was large (which implies a high service-level).

Some technical improvements were made in the Power Approximation by Mosier [1981], and the regression approach was used by Ehrhardt [1977] (compare Ehrhardt [1981]) to approximate the expected costs and other quantities of interest when the (s,S) policy is given.

There are other policies of a suboptimal form that have been used to control inventory systems, primarily because these policies are considerably easier to compute and analyze than are (s,S) policies. Two of the most popular are the (t,S) and (s,Q) policies (see Hadley and Whitin [1963, pp.235-295]). A (t,S) policy requires that every t periods an order be placed to raise the inventory position to S. An (s,Q) policy requires that whenever the inventory position falls below s, an order of size Q is to be placed. Using these policies Naddor [1975] developed rules for approximating the optimal (s,S) policy. These approximations were numerically compared in Kastner and Ehrhardt [1979] and in Ehrhardt and Kastner [1980]. The Power Approximation was always superior, although the Naddor Approximation did quite well for a large number of parameter settings.

We conclude by summarizing how these approximately optimal policies deal with the three difficulties described in Section 1.1. Typically, the renewal function is approximated using standard results from asymptotic renewal theory. Although we did not discuss it in detail in this section, the complicated total expected cost dependence on D is also dealt with using asymptotic renewal theory, with an empirically-based heuristic being implemented when D is small (in other words, the asymptotic assumptions are unreasonable). In order to have approximations that depend upon only a few moments of the demand distribution rather than the entire density, the form of the demand distribution is assumed to be given (typically Normal, Gamma, Poisson or Negative Binomial). The Power Approximation utilized a very general distribution form in which parameters were determined by a regression fit.

2. INVENTORY THEORY WITH SERVICE-LEVEL CONSTRAINTS

2.1 Model Formulation and Optimal Policies

A significant practical difficulty inherent in all these approximations (and exact algorithms) is the specification of shortage costs. These penalize backorders, but it can be very difficult to measure the cost impact of a backorder. It entails such things as "loss of goodwill" or "customer dissatisfaction", or equally difficult quantities to measure. One way to deal with this difficulty is to replace the shortage cost with a service-level constraint. For example, one can require that the frequency of backorders be no greater than 15%. Such measures can often be easier to specify than shortage costs.

A question of some interest is whether or not there is an optimal policy of the (s,S) form when a service-level constraint is used rather than a shortage cost. In general, there is not. Using a fairly general service-level constraint, Fromovitz [1965] showed that even in a single-period model the optimal policy may be randomized. He does show that if the service-level constraint is convex there is an optimal policy that is non-randomized, but most service-level constraints considered in the literature are not convex. Beesack [1967] has shown for the finite-horizon model (with the constraint that the expected number of stockouts be at most a prescribed fraction of expected demand) that (s_n,S_n) policies are still optimal. However, as most managers are unwilling to implement a policy more complicated than an (s,S) policy, most of the research is aimed at finding the best (s,S) policy that meets a given service-level.

2.2 Single-Item Models

Roberts [1962] showed that asymptotically the optimal value for D=S-s is independent of the shortage cost p, so typically it is assumed that D is given, and then approximations for s are derived such that the resulting (s,S) policy satisfies a prescribed service-level. The two service-levels most often considered are the one we use (frequency of backorders) and the one Beesack considered (fraction of expected demand met). In the spirit of Robert's work, Greenberg [1964] and Schneider [1978] derived closed form expressions for these two service-levels as functions of s and D. Using asymptotic renewal theory and both a Normal and Gamma demand distribution, Schneider [1978] derived approximations to find an approximately optimal s given D for both service-level constraints. Only the approximation to the constraint used by Beesack was investigated numerically. The Gamma Approximation always performed well, and the Normal Approximation performed well as long as the coefficient of variation was not too large. Tijms and Groenevelt [1982] extended these results to the case when the leadtime is a random variable.

2.3 Multi-Item Models

We now discuss the model whose study is the object of this paper. We consider an N-item inventory system in which each item i is an inventory model described in Section 1.1 without shortage costs assigned. Thus, for each item i,i=1,...,N, we have a fixed leadtime

k_i, fixed ordering cost K_i plus linear ordering cost c_i, and linear holding cost h_i. The demand realized for item i in any period has continuous cdf Φ_i, density ϕ_i, mean μ_i and finite standard deviation σ_i. All demands for all items in all periods are assumed to be independent. The criterion of optimality is the minimum expected undiscounted cost per period over an infinite horizon, and the service-level constraint to be satisfied in every period is

$$\sum_{i=1}^{N} W_i[\text{frequency of periods item i is not backordered}] \geq \alpha,$$

where $W_1,\ldots,W_N > 0$ and $0 < \alpha < 1$ are specified with $\sum_{i=1}^{N} W_i=1$.

A feasible policy is a pair (s,S), where $s=(s_1,\ldots,s_N)$ and $S=(S_1,\ldots,S_N)$, and the stationary policy (s_i,S_i) is followed for item i in every period.

This model has not been examined in the literature, but several related (but much simpler) models have. We briefly survey the results.

For a single-item inventory model with a shortage cost, if K=0 there is an optimal policy of the form (S,S), a base-stock policy (see Veinott [1966b]). Iglehart and Jaquette [1969] showed that if the shortage cost is replaced by a service-level constraint, a base-stock policy is still optimal. Mitchell [1982] considered this model with all leadtimes $k_i=0$ and exponential demands, and developed a computationally efficient algorithm to find the optimal base-stock policy. Evans [1967] considered a finite-horizon base-stock model in which total costs were minimized subject to a linear resource constraint. Using induction on the dynamic programming formulation, he derived the (extremely complicated) optimal policy.

In Section 1.2 we described suboptimal policy forms which, because of the relative ease in analyzing them, have been used in inventory control when $K_i>0$. Recall we described the (s,Q) policy (often called a (Q,r) policy in the literature), which requires that whenever the inventory position drops below s, an order of size Q is placed. There are several papers that consider our multi-item model (with perhaps a different constraint) governed by stationary (s,Q) rather than (s,S) policies.

There are two approaches to solving such problems that have been used, both of which are discussed in Hadley and Whitin [1963, pp.213-219, 304-307, 323-226]. One is to treat the problem as a constrained nonlinear program and then solve the first-order Kuhn-Tucker conditions. The second is to formulate the model as a dynamic program (a generalized Knapsack Problem) and then solve it stage by stage.

The method using the Kuhn-Tucker conditions is used in Winters [1962], Parker [1964], Gerson and Brown [1970], Presutti and Trepp [1970], Schrady and Choe [1971], and Schroeder [1974]. The method using dynamic programming is used in Kaplan [1978].

Winters [1962] minimized total expected cost subject to a production smoothing constraint. Parker [1964] minimized total expected cost subject to a service-level constraint. Gerson and Brown [1970] minimized total expected cost subject to a service-level constraint and s=0. Presutti and Trepp [1970] also minimized total expected cost subject to a service-level constraint. Schrady and Choe [1971] maximized a service-level criterion subject to investment budget constraints, as did Schroeder [1974]. Kaplan [1972] considered a somewhat different model, a finite-horizon model with budget constraints that penalize dollars spent before the end of the horizon. It is a single-item model with s=0. We note that the models in Parker [1964] and Schroeder [1974] assume a continuous rather than periodic review of the inventory position. Such models tend to produce policies that give rather poor performance in a periodic-review setting (see Wagner et al [1965]).

For two reasons, all of these studies are somewhat unsatisfactory. First, as mentioned, (s,Q) policies are not optimal. Second, in none of them are the policies produced compared with any other method, making it difficult to evaluate the performance of such policies in a multi-item inventory system. In our paper we use (s,S) policies and derive an algorithm, the GKD (Generalized Knapsack Duality) Algorithm to approximately solve our model, and then compare the (s,S) policies produced with an approach frequently used in practice by managers, the Identical Service Approach (compare Mitchell [1982]). This approach sets shortage costs for each item in an inventory system so that the service-level for each item is the same, say α. Then each item is treated individually, with policies being computed by some existing method. We will show that using our model, which in effect varies the individual item service-levels while still maintaining an overall service-level of α (via our service-level constraint), total expected costs can be significantly reduced for many multi-item inventory systems. This is the first issue with which we deal.

The second issue involves aggregating very large inventory systems in such a way that a representative fraction of them can be used to determine the policies for all of them, which we discuss next.

3. MULTI-ITEM INVENTORY SYSTEM AGGREGATION

Many real-world inventory systems contain thousands of items. It would be cost prohibitive to try to include every one of them in any model which is to be used to set inventory policies. Inventory aggregation is any method of partitioning such a system into blocks, each of which will be dealt with essentially the same. For example, a representative item could be chosen from each block to determine the policy that will be used for all items in the block. Thus the problem is reduced to more manageable proportions. The question is, of course, how to aggregate so that the resulting policies are as close as possible (in a total expected cost sense) to policies that would have been produced had every item in the inventory system been included in the model.

Very little is known about this problem (see Bitran and Hax [1982], which deals with a finite horizon, deterministic demand inventory system). However, extensive empirical investigations of many real-world inventory systems have suggested some remarkable similarities in the overall distribution of items in inventory systems. These results are discussed in Brown [1959], [1963], [1967] and [1977], and Peterson and Silver [1979, pp. 30-37, 71-80]. We summarize the observations in Peterson and Silver.

The annual dollar demand of an item is defined to be its annual expected demand times its value per unit. Of course, the "value" of an item may be difficult to determine exactly, but can often be taken to be its ordering or holding cost. Typically, five to ten percent of the items in an inventory system account for about fifty percent of the total annual dollar demand of the entire inventory.

These items tend to be very important, and are denoted Type A items. On the other hand, typically about thirty to fifty percent of the items account for only five to ten percent of the total annual dollar demand. These are the relatively unimportant items, and are denoted Type C items. The rest of the items in the inventory system, the moderately important ones, are denoted Type B items.

If the items in such an inventory system are ordered in descending order by their annual dollar demand, and then the cumulative percent of the items is plotted against the corresponding cumulative percent of annual dollar demand, the graph would look something like Exhibit 1. Notice also that about twenty percent of the items account for about eighty percent of the total annual dollar demand. This also is typical for many real-world inventory systems.

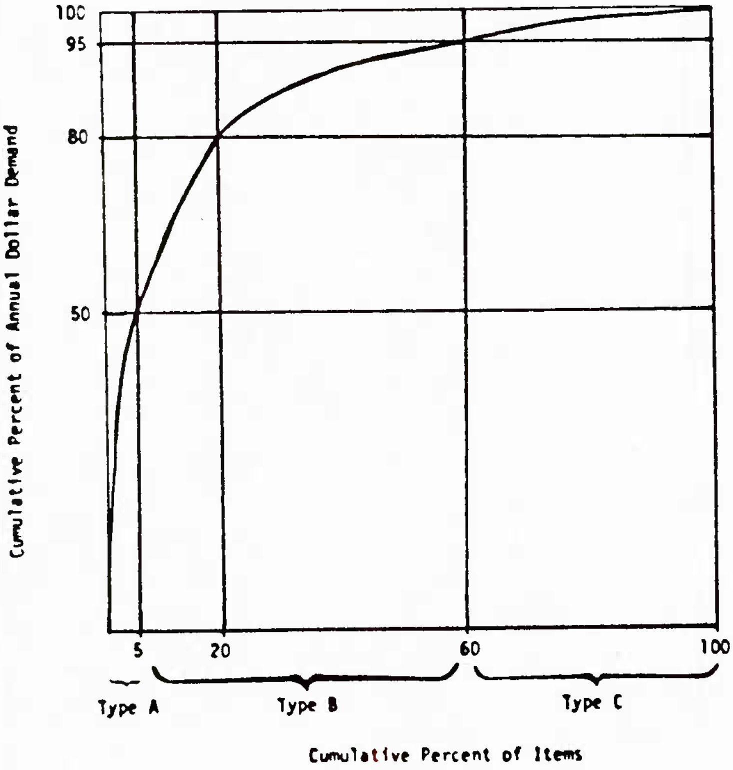


Exhibit 1

Empirical observations have shown that this graph can very often be well-approximated by a lognormal cumulative distribution function.

We address the second issue of this paper, large-scale inventory aggregation, by showing that sampling techniques which stratify on the item types and which maintain the structure illustrated in Figure 1 can be used to accurately predict the (s,S) policies for all the items in the inventory system. We exploit the special structure of our inventory problem by using the sample to estimate the Lagrange constraint multiplier of the inventory problem, from which we compute all (s,S) policies.

II. THE GENERALIZED KNAPSACK DUALITY (GKD) ALGORITHM

1. INTRODUCTION

We consider a multi-item inventory system and study the problem of specifying both system-wide and individual service objectives. Most theoretical models considered in the literature assume that holding or shortage costs are applied to any excess inventory or unsatisfied demand, respectively. A major difficulty in applying such models in practice is the specification of shortage costs. Frequently, the manager sets shortage cost parameters based on an objective of satisfying demand with at least some minimum probability. Such an approach may be preferred because subjective factors can more easily be expressed as a probability of satisfying demand than as a cost for each shortage incurred.

A significant shortcoming of this approach is that it usually entails the setting of a probability of demand satisfaction that is applied uniformly to all items in the system. In this paper we raise the issue of specifying different service objectives for individual items while still satisfying some given system-wide objective. The value of this approach is that total system costs can be reduced below those of the method that requires identical service for all items.

We are interested in the following multi-item inventory system. For each item i there is a set-up ordering cost K_i, a unit ordering cost c_i, and a fixed leadtime k_i between placement and delivery of orders. These costs are assessed upon order delivery. There is a unit holding cost h_i assessed at the end of each period, and the demand realized for for item i in any period has absolutely continuous cumulative distribution function (cdf) Φ_i, density ϕ_i, mean μ_i, and finite standard deviation σ_i. All demands for all items in all periods are mutually independent. There are N items, and the criterion of optimality is the minimum expected undiscounted cost per period over an infinite horizon. We do not include shortage costs to penalize back-orders. Rather we introduce the following service-level constraint to be satisfied in every period:

$$\sum_{i=1}^{N} W_i[\text{frequency of periods item i is not backordered}] \geq \alpha,$$

where $0<\alpha<1$, $W_1,\ldots,W_N>0$ and $\sum_{i=1}^{N} W_i=1$.

For such an inventory model, an (s,S) policy requires that whenever the inventory position (inventory on hand plus on order) drops below s, an order is placed bringing the inventory position up to S. An (s,S) policy is called stationary if it is applied in every period. We only consider stationary (s,S) policies because when shortage costs are included in the model, there is an optimal policy which is stationary and (s,S) (Iglehart [1963]). Any stationary (s,S) policy for our model gives rise to a unique and well-defined specification of shortage costs. If we denote by $P_i(s_i,S_i)$ the frequency of periods that item i is not backordered when following the policy (s_i,S_i), then, when ϕ_i is continuous,

$$P_i(s_i,S_i) = \frac{p_i}{p_i+h_i}, \qquad (1)$$

where p_i is the shortage cost under which the policy (s_i,S_i) is optimal

(Veinott and Wagner [1965]). We can solve this equation for p_i and thus determine the shortage costs associated with any stationary (s,S) policy.

For our model a feasible policy is a set $\{(s,S)\}$, where $s=(s_1,\ldots,s_N)$ and $S=(S_1,\ldots,S_N)$. For item i we follow the policy (s_i,S_i) in every period. We define D_i by $D_i=S_i-s_i$, and denote by Φ_i^* the (k_i+1)-fold convolution of Φ_i with itself, and by ϕ_i^* its density. Also let μ_i^* and σ_i^* be the mean and standard deviation of a random variable with cdf Φ_i^*. In Mitchell [1982] we considered this problem when $K_i=0$ for all items i, and developed a computationally efficient algorithm for the case when all $k_i=0$ and all demands are exponentially distributed. We extend those results in this paper, dropping all three of these restrictions.

Section 2 describes the formulation of our inventory problem as a non linear program. Using the Kuhn-Tucker conditions, we prove necessary and sufficient conditions for (possibly local) optimality. The main results are found in Theorems 1, 2, 6, 7 and 8 (the reader may choose to skip the proofs of these results). There is at the end of this chapter a list of the notation, a summary of frequently used relations, and a summary of the theorems and lemmas.

Section 3 is a description of the algorithm. It is based on local duality theory of nonlinear programming, and the proof of convergence uses results from Section 2. Section 4 reports a numerical investigation. In Section 5 we discuss our recommendations when the sufficient conditions may not hold for some items in the inventory system. Finally, Section 6 contains some conclusions.

2. ANALYSIS

If we denote by $C_i(S_i,D_i)$ the expected average undiscounted cost per period over an infinite horizon, and by $P_i(S_i,D_i)$ the frequency of periods that item i is not backordered, we have the following result.

Theorem 1. For $S_i \geq D_i$,

$$\text{(i)} \quad C_i(S_i,D_i) = \frac{1}{1+M_i(D_i)} \left\{ h_i \left[\int_0^{D_i} \int_{-\infty}^{S_i-y} (S_i-y-x)\phi_i{}^*(x)m_i(y)dxdy + \int_{-\infty}^{S_i} (S_i-x)\phi_i{}^*(x)dx \right] + K_i \right\}$$

$$\text{(ii)} \quad P_i(S_i,D_i) = \frac{1}{1+M_i(D_i)} \left\{ \int_0^{D_i} \Phi_i{}^*(S_i-y)m_i(y)dy + \Phi_i{}^*(S_i) \right\},$$

where M_i is the renewal function of Φ_i and $m_i = M_i'$.

Proof. See Roberts [1962], Greenberg [1964] and Ehrhardt [1981] □.

For two reasons we restrict attention to (s_i,S_i) policies with $s_i \geq 0$ (equivalently, $S_i \geq D_i$). First, the expressions given in Theorem 1 do not hold when $s_i < 0$. One must replace with $(0,S_i)$ the limits of integration in the first integral of C_i and the limits of integration in P_i. Second, many managers consider policies with $s_i < 0$ to be unsatisfactory for practical implementation.

Thus we want to solve the following constrained nonlinear program (NLP) to obtain an optimal policy for our inventory system:

$$\text{minimize } C(S,D) = \sum_{i=1}^{N} C_i(S_i,D_i)$$

$$\text{subject to } P(S,D) = \sum_{i=1}^{N} W_i\left[P_i(S_i,D_i)-\alpha\right] \geq 0$$

$$S_i \geq D_i,\ i=1,\ldots,N \quad ,$$

where $0<\alpha<1$, $W_1,\ldots,W_N>0$ and $\sum_{i=1}^{N} W_i=1$.

There are three serious difficulties in finding a solution. First, the renewal function M_i generally does not have a closed-form expression, and usually is complicated when it does. Second, the functional dependence of C_i and P_i on the variable D_i is complicated, as is easily seen if one works out the partial derivatives. Third, the behavior of the functions depends heavily on the demand distributions Φ_i.

We deal with the first difficulty by using asymptotic approximations for the renewal functions. In particular, Smith [1954] has shown

$$M_i(y) = \frac{y}{\mu_i} + \frac{\sigma_i^2}{2\mu_i^2} - \frac{1}{2} + o(1) \text{ as } y\to\infty. \tag{2}$$

We use the following approximations.

$$M_i(D_i) \doteq \frac{D_i}{\mu_i} + \frac{\sigma_i^2}{2\mu_i^2} - \frac{1}{2}.$$

$$\frac{1}{1+M_i(D_i)} \doteq \frac{\mu_i}{D_i+(\mu_i+\sigma_i^2/\mu_i)/2} \equiv \rho_i. \tag{3}$$

We approximate $m_i(y)=M_i'(y)$ by a constant function, since $M_i(y)$ is asymptotically affine. One could differentiate (2) and approximate $m_i(y)$ by $\frac{1}{\mu_i}$, but we use the somewhat better approximation discussed in Ehrhardt [1981], namely,

$$m_i(y) \doteq \frac{1-\rho_i}{\rho_i D_i}, \quad y\varepsilon(s_i,S_i). \qquad (4)$$

(This approximation approaches $\frac{1}{\mu}$ as $D_i \to \infty$).

For the complicated functional dependence on D_i, we approximate D_i by a constant, and regard C_i and P_i as functions only of S_i. Roberts [1962] proves that in the single-item case where a shortage cost p is specified, the optimal value D* of D is given by

$$D^* = \sqrt{\frac{2K\mu}{h}} + \text{constant} + o(1)$$

as $K \to \infty$, $p \to \infty$ such that $\frac{K}{p}$ remains constant.

Since D* is asymptotically independent of the shortage cost p, and hence by (1) of the service-level α, it would seem reasonable to approximate D* by the constant $\sqrt{\frac{2K\mu}{h}}$, the so called "economic order quantity."

An extensive empirical comparison of approximately optimal inventory policies by Wagner, et al [1965], showed this to be a remarkably good approximation, even for K and p fairly small, although it seems to give better results when $\sqrt{\frac{2K\mu}{h}} \geq 1.5\mu$. Moreover, the total expected cost is typically quite flat near D*, so if S* is well-approximated, the approximation for D* does not need to be especially accurate to have a total cost close to the cost associated with the optimal policy.

Actually, we use the Power Approximation discussed in Mosier [1981] for the optimal D. He reports empirical investigations that indicate that this approximation is superior to the economic order quantity. As in the case of the economic order quantity, this approximation tends to be poor when it is smaller than 1.5μ. For the derivation of our algorithm, we assume D_i is at least μ_i, and thus use the approximation

$$D_i = \max\left\{\mu_i,\ 1.3\mu_i^{.494}\left(\frac{K_i}{h_i}\right)^{.506}\left[1+\left(\frac{\sigma_i^*}{\mu_i}\right)^2\right]^{.116}\right\} . \qquad (5)$$

We note that asymptotically (as $\frac{K_i}{h_i} \to \infty$) this approximation differs very little from the economic order quantity.

To deal with the third difficulty, we intended to assume that the demand distribution in Normal, since Wagner, et al [1965], Wagner [1975, ch. 19], MacCormick [1974], Estey and Kauffman [1975], MacCormick et al [1977], and MacCormick [1977] all suggest that a Normal distribution gives good approximations to the optimal policies even when the true distribution is skewed (By a good approximation we mean that the expected total cost of the approximation policy is nearly the same as for an optimal policy). But because the Normal cdf cannot be expressed in a closed form, we use a Logistic distribution to approximate the Normal distribution. It is an especially simple and accurate approximation to the Normal distribution, as discussed in Johnson and Kotz [1970, ch. 22].

They show that if Φ^* is the cdf of a Normal random variable with mean μ^* and standard deviation σ^*, then

$$\Phi^*(x) \doteq \frac{1}{1+\exp(-\gamma[x-\mu^*])}, \tag{6}$$

where

$$\gamma = \frac{15\pi}{16\sqrt{3}\sigma^*}.$$

This is the standard Logistic distribution with parameters 0 and $\frac{1}{\gamma}$.

We introduce the change of variables

$$\left.\begin{aligned} \theta_i &= e^{-\gamma_i(S_i-\mu_i^*)}, \\ \text{where} \qquad & \\ \gamma_i &= \frac{15\pi}{16\sqrt{3}\,\sigma_i^*}, \end{aligned}\right\} \tag{7}$$

and reformulate the NLP using the approximations discussed. The result is given in Theorem 2 and (17). Since the transformation

$$\theta = (\theta_1,\ldots,\theta_N)$$

is one-to-one, infinitely differentiable,and invertible everywhere, the problem as formulated in θ is equivalent to the problem as formulated in S. We note that $\theta_i>0$ for any finite S_i. It is clear from the formulation of the problem that the optimal S is finite, and so the optimal θ is positive. Thus we do not need to add the constraint $\theta_i\geq 0$.

Theorem 2. If the approximations (2) through (6) and the change of variables (7) are used in the expressions for $C_i(S_i,D_i)$ and $P_i(S_i,D_i)$, and if we denote by $C_i(\theta_i)$ and $P_i(\theta_i)$ the resulting approximations (without policy-independent terms and a function in C_i that is $o(1)$ as

$S_i \to \infty$), we have

$$\text{(i)} \quad C_i(\theta_i) = \frac{h_i}{\gamma_i} \log \left[\frac{(1+\theta_i)^{\rho_i}}{\theta_i}\right]$$

$$\text{(ii)} \quad P_i(\theta_i) = (1-\rho_i)\left[1 + \frac{1}{\gamma_i D_i} \log\left(\frac{1+\theta_i}{1+\delta_i\theta_i}\right)\right] + \frac{\rho_i}{1+\theta_i} \ ,$$

where $\delta_i = e^{\gamma_i D_i}$.

Proof. We suppress the subscript i.

(i) If the approximations (2) through (5) are used in the expression for $C(S,D)$, and $C(S)$ denotes the resulting approximation without policy-independent terms, then we have

$$C(S) = \rho\left[h\int_0^D \int_{-\infty}^{S-y} (S-y-x)\phi^*(x)\frac{1-\rho}{\rho D}\, dxdy + h\int_{-\infty}^{S} (S-x)\phi^*(x)dx\right]$$

$$- \text{ (policy-independent terms).}$$

Now

$$\int_0^D \int_{-\infty}^{S-y} (S-y-x)\phi^*(x)dxdy$$

$$= \int_0^D \int_{-\infty}^{\infty} (S-y-x)\phi^*(x)dxdy - \int_0^D \int_{S-y}^{\infty} (S-y-x)\phi^*(x)dxdy, \quad (8)$$

provided these integrals exist.

We show that they exist by evaluating them. First,

$$\int_0^D \int_{-\infty}^{\infty} (S-y-x)\phi^*(x)dxdy = \int_0^D (S-y)\left[\int_{-\infty}^{\infty} \phi^*(x)dx\right]dy - \int_0^D \left[\int_{-\infty}^{\infty} x\phi^*(x)dx\right]dy$$

$$= \int_0^D (S-y)dy - \int_0^D \mu^* dy$$

$$= DS - D\left(\mu^* + \frac{D}{2}\right).$$

To evaluate the other integral in (8) we define

$$\varepsilon(S) = -\int_0^D \int_{S-y}^{\infty} (S-y-x)\phi^*(x)dxdy$$
$$= \int_0^D \int_{S-y}^{\infty} (x+y-S)\phi^*(x)dxdy.$$

It is straightforward to verify that on the domain of integration

$$0 \leq (x+y-S)\phi^*(x) \leq x\phi^*(x).$$

Since $\mu^* < \infty$ and $D < \infty$, we have

$$\int_0^D \int_{-\infty}^{\infty} |x\phi^*(x)|dxdy < \infty,$$

and so the Dominated Convergence Theorem (Hoffman [1975,p. 331]) justifies taking the derivative of ε under the integral. This gives

$$\varepsilon'(S) = -\int_0^D \int_{S-y}^{\infty} \phi^*(x)dxdy$$
$$= -\int_0^D [1-\Phi^*(S-y)]dy$$
$$\geq -D \max_{0 \leq y \leq D} \{1-\Phi^*(S-y)\}$$
$$= -D\,[1-\Phi^*(S-D)].$$

Obviously $\varepsilon'(S) \leq 0$, so

$$0 \geq \lim_{S\to\infty} \varepsilon'(S) \geq \lim_{S\to\infty} \{-D[1-\Phi^*(S-D)]\} = 0,$$

which implies $\lim_{S\to\infty} \varepsilon'(S) = 0$. Thus we have $\varepsilon(S) = a+o(1)$ as $S \to \infty$, for some constant a.

By assumption we do not include the o(1) function in the approximate cost function, and so we have that

$$C(S) = h(1-\rho)S + h\rho\int_{-\infty}^{S}(S-x)\phi^*(x)dx, \tag{9}$$

where we have dropped all policy-independent terms. Using the approximation (6), we have that

$$\int_{-\infty}^{S}(S-x)\phi^*(x)dx = \int_{-\infty}^{S}\frac{(S-x)\,\gamma\, e^{-\gamma(x-\mu^*)}}{\left[1+e^{-\gamma(x-\mu^*)}\right]^2}\,dx.$$

After the change of variables $X=e^{-\gamma(x-\mu^*)}$, this becomes

$$\int_{\theta}^{\infty}\frac{(S-\mu^*)+\frac{1}{\gamma}\log X}{(1+X)^2}\,dX, \tag{10}$$

where θ is given by (7). Then

$$\int_{\theta}^{\infty}\frac{dX}{(1+X)^2} = \frac{1}{1+\theta} \tag{11}$$

and

$$\int_{\theta}^{\infty}\frac{\log X}{(1+X)^2}dX = \frac{-\log\theta}{1+\theta} + \log\theta - \log(1+\theta), \tag{12}$$

and so (7), (9), (10), (11), (12) and the definition of $C(\theta)$ give

$$\begin{aligned}C(\theta) &= h(1-\rho)(\mu^* - \frac{1}{\gamma}\log\theta)\\ &\quad - \frac{h\rho}{\gamma}\log + \frac{h\rho}{\gamma}\log(1+\theta)\\ &\quad - (\text{policy-independent terms})\\ &= \frac{h\rho}{\gamma}\log(1+\theta) - \frac{h}{\gamma}\log\theta\\ &= \frac{h}{\gamma}\log\left[\frac{(1+\theta)^{\rho}}{\theta}\right].\end{aligned}$$

(ii) If the approximations (2) through (5) are used in the expression for P(S,D), and P(S) denotes the resulting approximation, then we have

$$P(S) = \rho\left\{\int_0^D \Phi^*(S-y)\ \frac{1-\rho}{\rho D}\,dy + \Phi^*(S)\right\}$$

$$= \frac{1-\rho}{D}\int_0^D \Phi^*(S-y)dy + \rho\Phi^*(S). \qquad (13)$$

Using approximation (6) we have

$$\int_0^D \Phi^*(S-y)dy = \int_0^D \frac{dy}{1-e^{-\gamma(S-y-\mu^*)}} \ .$$

Making the change of variables $Y = e^{-\gamma(S-y-\mu^*)}$, this becomes

$$\frac{1}{\gamma}\int_{-\gamma(S-\mu^*)}^{-\gamma(S-\mu^*)} \frac{dY}{1+e^Y} \ . \qquad (14)$$

Then

$$\int \frac{dY}{1+e^Y} = Y - \log(1+e^Y), \qquad (15)$$

so (7), (13), (14), (15) and the definition of $P(\theta)$ give

$$P(\theta) = \frac{1-\rho}{D\gamma}\left[\gamma D + \log\left(\frac{1+\theta}{1+\delta\theta}\right)\right] + \frac{\rho}{1+\theta}$$

$$= (1-p)\left[1 + \frac{1}{\gamma D}\log\left(\frac{1+\theta}{1+\delta\theta}\right)\right] + \frac{\rho}{1+\theta}\ \square.$$

Let

$$\theta_i^u = e^{-\gamma_i(D_i-\mu_i^*)}, \qquad (16)$$

so $S_i \geq D_i$ is equivalent to $\theta_i \leq \theta_i^u$. Then we have formulated the following NLP, the solution to which gives an approximately optimal policy for our multi-item inventory system via the change of variables (7):

$$\left.\begin{aligned} &\text{minimize } C(\theta) = \sum_{i=1}^{N} C_i(\theta_i) \\ &\text{subject to } P(\theta) = \sum_{i=1}^{N} W_i[P_i(\theta_i)-\alpha] \geq 0, \\ &\theta_i \leq \theta_i^{u} \;, i=1,\ldots,N. \end{aligned}\right\} \qquad (17)$$

We use nonlinear programming techniques to solve (17). Specifically, we derive first-order necessary conditions for optimality using the familiar Kuhn-Tucker conditions. Although the objective function is convex, the feasible region in general is not. Therefore the necessary conditions may not be sufficient to guarantee even local optimality (Luenberger [1973, Sections 6.4, 10.6]). We show that for D_i sufficiently large, however, the Kuhn-Tucker conditions are sufficient for (at least) local optimality. We also describe a computationally efficient algorithm for finding an optimal policy when it exists, and compare the performance of these policies with policies generated by an existing method.

The following formulas are needed.

Lemma 3. For i=1,...,N,

(a) $$C_i'(\theta_i) = -\frac{h_i}{\gamma_i}\left[\frac{1+(1-\rho_i)\theta_i}{(1+\theta_i)\theta_i}\right]$$

(b) $$C_i''(\theta_i) = \frac{h_i}{\gamma_i}\left[\frac{(1-\rho_i)\theta_i^2 + 2\theta_i + 1}{(1+\theta_i)^2\theta_i^2}\right]$$

(c) $$P_i'(\theta_i) = -\frac{[D_i\rho_i\gamma_i + (1-\rho_i)(\delta_i-1))] + [D_i\rho_i\gamma_i\delta_i + (1-\rho_i)(\delta_i-1)]\theta_i}{\gamma_i D_i(1+\theta_i)^2(1+\delta_i\theta_i)} .$$

Proof (a) Differentiating $C_i(\theta_i)$ and suppressing i gives,

$$C'(\theta) = \frac{h}{\gamma}\left[\frac{\rho}{1+\theta} - \frac{1}{\theta}\right] = -\frac{h}{\gamma}\left[\frac{1+(1-\rho)\theta}{(1+\theta)\theta}\right]. \tag{18}$$

(b) Differentiating (18) gives

$$C''(\theta) = \frac{h}{\gamma}\left[\frac{\rho}{-(1+\theta)^2} + \frac{1}{\theta^2}\right] = \frac{h}{\gamma}\left[\frac{(1-\rho)^2+2\theta+1}{(1+\theta)^2\theta^2}\right].$$

(c) Differentiating $P_i(\theta_i)$ and supressing i gives,

$$P'(\theta) = \frac{(1-\rho)}{\gamma D}\left[\frac{1}{1+\theta} - \frac{\delta}{1+\delta\theta}\right] - \frac{\rho}{(1+\theta)^2} \tag{19}$$

$$= \frac{(1-\rho)(1-\delta)}{\gamma D(1+\theta)(1+\delta\theta)} - \frac{\rho}{(1+\theta)^2}.$$

$$= -\frac{(1-\rho)(\delta-1)(1+\theta) + D\rho\gamma(1+\delta\theta)}{\gamma D(1+\theta)^2(1+\delta\theta)}$$

$$= -\frac{[D\rho\gamma + (1-\rho)(\delta-1)] + [D\rho\gamma\delta + (1-\rho)(\delta-1)]\theta}{\gamma D(1+\theta)^2(1+\delta\theta)} \square.$$

Theorem 4. Suppose

$$P(\theta^u) \geq 0. \tag{20}$$

Then $\theta^* = \theta^u$ is globally optimal for (17).

Proof. It is straightforward to verify (since ρ_i is decreasing as a function of D_i) that (making use of (5))

$$D_i \geq \mu_i \text{ implies } 0 < \rho_i < \frac{2}{3}. \tag{21}$$

This with Lemma 3(a) implies that

$$C_i'(\theta_i) < 0 \text{ whenever } \theta_i \leq \theta_i^u, \tag{22}$$

and so $C(\boldsymbol{\theta})$ is decreasing in each argument for $\boldsymbol{\theta}_i \le \boldsymbol{\theta}_i^u$. But by hypothesis $\boldsymbol{\theta}=\boldsymbol{\theta}^u$ is feasible, so $\boldsymbol{\theta}=\boldsymbol{\theta}^u$ is a global minimum of (17) $\square$.

In order to investigate the case when $P(\theta^u)<0$, we make use of the Kuhn-Tucker necessary and sufficient conditions for optimality. To use these, however, one must verify that some constraint qualification holds at the optimal point. We show that if $P(\theta^u)<0$, then every feasible point is a regular point, that is, the gradients of the active constraints at the point are linearly-independent. Luenberger [1973, 223-227] shows that this constraint qualification is sufficient to guarantee that if the Kuhn-Tucker necessary and sufficient conditions hold at a point,then that point is an (at least local) optimum.

Lemma 5. If $P(\boldsymbol{\theta}^u)<0$, then any $\boldsymbol{\theta}$ feasible for (17) is a regular point of (17), that is, the gradients (at θ) of the constraints active at θ are all linearly-independent.

Proof. Let θ be feasible for (17). Without loss of generality, suppose that of the constraints $\theta_i \le \theta_i^u$, $i=1,\ldots,N$, the first j are active and the last N-j are inactive. Since by assumption $\theta=\theta^u$ is not feasible, we have $N-j \ge 1$. Let T denote the matrix which has as its columns the gradients (at θ) of the constraints active at θ. If the constraint $P(\theta)\ge 0$ is inactive, then

$$T = \begin{array}{c|ccc|} & & j & \\ \hline & 1 & & 0 \\ j & & \ddots & \\ & 0 & & 1 \\ \hdashline N-j & 0 & \cdots & 0 \\ \hline \end{array},$$

which certainly has linearly-independent columns. If the constraint $P(\boldsymbol{\theta})\ge 0$ is active, then

$$T = \begin{array}{cc} & \begin{array}{cc} \qquad j \qquad & \qquad 1 \qquad \end{array} \\ \begin{array}{c} j \\ \\ \\ N-j \\ \\ \end{array} & \left[\begin{array}{ccc|c} 1 & & 0 & W_1P_1'(\theta_1) \\ & \ddots & & W_2P_2'(\theta_2) \\ 0 & & 1 & \cdot \\ \hline & & & \vdots \\ 0 & \cdots & 0 & W_NP_N'(\theta_N) \end{array}\right] \end{array} .$$

Certainly the first j columns of T are linearly-independent. If the last column were a linear combination of the first j, then (since $N-j \geq 1$) $W_NP_N'(\theta_N)=0$. Now it is straightforward to verify that

$$D_i \geq 0 \text{ implies } \delta_i > 1, \tag{23}$$

so this with Lemma 3(c) and (21) implies that $P_N'(\theta_N)>0$. And since $W_N > 0$, we have a contradiction, and so the columns of T are linearly-independent, which implies that θ is a regular point of (17) □.

Theorem 6. Suppose $P(\theta^U)<0$ and for $i=1,\ldots,N$, let

$$\begin{aligned}
A_i &= A_i(D_i) = D_ih_i \\
B_i &= B_i(D_i) = D_ih_i[2+\delta_i-\rho_i] \\
E_i &= E_i(D_i) = D_ih_i[\delta_i+(1-\rho_i)(\delta_i+1)] \\
F_i &= F_i(D_i) = D_ih_i(1-\rho_i)\delta_i \\
G_i &= G_i(D_i) = W_i[D_i\rho_i\gamma_i + (1-\rho_i)(\delta_i-1)] \\
H_i &= H_i(D_i) = W_i[D_i\rho_i\gamma_i\delta_i + (1-\rho_i)(\delta_i-1)]
\end{aligned}$$

$$\lambda_i(\theta_i) = \frac{A_i+B_i\theta_i+E_i\theta_i^2+F_i\theta_i^3}{G_i\theta_i+H_i\theta_i^2} .$$

Then the first-order Kuhn-Tucker necessary condition that θ^* be a minimum of (17) is the following:

There exists a real number $\lambda^*\geq 0$ such that

$$\begin{aligned} &(a)\quad \lambda_i(\theta_i^*) = \lambda^* \quad \text{when } \theta_i^*<\theta_i^u \\ &\qquad\ \lambda_i(\theta_i^*) \geq \lambda^* \quad \text{when } \theta_i^*=\theta_i^u \\ &(b)\qquad P(\theta^*) = 0. \end{aligned} \tag{24}$$

Proof. As proved in Luenberger [1973, pp.232-234], given the constraint qualification guaranteed in Lemma 5, a first-order necessary condition for θ^* to be a minimum of (17) is that θ^* be feasible for (17), and that there exists $\lambda^*\geq 0$ and $\boldsymbol{\zeta}^* = (\zeta_1^*,\ldots,\zeta_N^*) \geq 0$ such that

$$\left.\begin{aligned} \nabla L(\boldsymbol{\theta}^*,\boldsymbol{\lambda}^*,\boldsymbol{\zeta}^*) \equiv \nabla C(\boldsymbol{\theta}^*) - \lambda^*\nabla P(\boldsymbol{\theta}^*) - \zeta^* I = 0 \\ \lambda^* P(\boldsymbol{\theta}^*) = 0 \\ \zeta^{*T}(\boldsymbol{\theta}^*-\boldsymbol{\theta}^u) = 0, \end{aligned}\right\} \tag{25}$$

where I is the NxN identity matrix, and a superscript T denotes transpose.

By assumption $P(\theta^u)<0$, and clearly

$$P(0) = \sum_{i=1}^{N} W_i(1-\alpha) > 0; \tag{26}$$

since it cannot cost less to give a higher service-level, we can assume that at optimality $P(\theta^*) = 0$. Now Lemma 3(c), (21) and (23) imply that for all i,

$$P_i'(\boldsymbol{\theta}_i)<0 \text{ whenever } \boldsymbol{\theta}_i\leq\boldsymbol{\theta}_i^u, \tag{27}$$

and so by (22) we have that (25) is equivalent to the following condition:

There exists $\lambda^*\geq 0$ such that for all i,

$$\left.\begin{array}{ll}\text{(a)} & \dfrac{C_i'(\theta_i^*)}{W_iP_i'(\boldsymbol{\theta}_i^*)} = \lambda^* \quad \text{when} \quad \theta_i^* < \theta_i^u \\[2ex] & \dfrac{C_i'(\boldsymbol{\theta}_i^*)}{W_iP_i'(\boldsymbol{\theta}_i^*)} \geq \lambda^* \quad \text{when} \quad \theta_i^* = \theta_i^u \\[2ex] \text{(b)} & P(\boldsymbol{\theta}^*) = 0. \end{array}\right\} \qquad (28)$$

We note that $\zeta_i^* = C_i'(\boldsymbol{\theta}_i^*) - \lambda^* P_i'(\boldsymbol{\theta}_i^*)$, for each i. To complete the proof, we need only to show

$$\lambda_i(\theta_i^*) = \frac{C_i'(\theta_i^*)}{W_iP_i'(\theta_i^*)} .$$

We suppress the subscript i. By Lemma 3(a) and (c),

$$\frac{C'(\theta)}{WP'(\theta)} = \left(-\frac{h}{\gamma}\right)\left[\frac{1+(1-\rho)\theta}{(1+\boldsymbol{\theta})\theta}\right](-\gamma D)\left[\frac{(1+\boldsymbol{\theta})^2(1+\boldsymbol{\delta\theta})}{G+H\boldsymbol{\theta}}\right].$$

After factoring (1+θ) out of the numerator and denominator, the numerator becomes $Dh[1+(1-\rho)\boldsymbol{\theta}][1+(\boldsymbol{\delta}+1)\boldsymbol{\theta}+\boldsymbol{\delta\theta}^2]$, which, when multiplied out, is $A+B\boldsymbol{\theta}+E\boldsymbol{\theta}^2+F\boldsymbol{\theta}^3$, and the denominator is $G\boldsymbol{\theta}+H\boldsymbol{\theta}^2$. Thus we have

$$\frac{C_i'(\boldsymbol{\theta}_i)}{W_iP_i'(\boldsymbol{\theta}_i)} = \lambda_i(\theta_i), \qquad (29)$$

as was to be shown □.

Although the first-order Kuhn-Tucker condition is necessary for (at least local) optimality, in general it is not sufficient. In Theorem 7 we give a condition sufficient to guarantee that the Hessian of the Lagrangian of (17) is positive-definite, a strong second-order sufficient condition (see Luenberger [1973, p.235]). In Theorem 8 we give conditions in terms of D_i which guarantee that the hypotheses of Theorem 7 are satisfied.

Theorem 7. Suppose $P(\theta^u)<0$ and that there exists $\lambda^*\geq 0$ so that the necessary conditions (24) hold at θ^*. If also $\lambda_i(\theta_i)$ is decreasing for $\theta_i \le \theta_i^u$, then the Hessian matrix $\nabla_{\boldsymbol{\theta\theta}} L(\boldsymbol{\theta}^*, \lambda^*, \zeta^*)$ is positive-definite, which implies that (24) is sufficient for (at least local) optimality.

Proof. We first note that Lemma 3(b) and (21) imply that

$$C_i''(\boldsymbol{\theta}_i) > 0. \tag{30}$$

We have

$$\nabla_{\boldsymbol{\theta\theta}} L(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*, \boldsymbol{\zeta}^*) = \nabla^2 C(\boldsymbol{\theta}^*) - \lambda^* \nabla^2 P(\boldsymbol{\theta}^*)$$

$$= \begin{bmatrix} C_1''(\boldsymbol{\theta}_i^*) - \lambda^* W_1 P_1''(\boldsymbol{\theta}_i^*) & & 0 \\ & \ddots & \\ 0 & & C_N''(\theta_N^*) - \lambda^* W_N P_N''(\theta_N^*) \end{bmatrix}.$$

Let Q_i denote the i^{th} diagonal entry of $\nabla_{\boldsymbol{\theta\theta}} L$. We show that $Q_i > 0$ for each i, as this implies that $\nabla_{\boldsymbol{\theta\theta}} L$ is positive-definite. If $P_i''(\theta_i^*) \le 0$, (30) implies $Q_i > 0$, since λ^*, $W_i > 0$. If $P_i''(\theta_i^*) > 0$, (24) implies $Q_i \ge C_i''(\theta_i^*) - [\lambda_i(\boldsymbol{\theta}_i^*)] W_i P_i''(\theta_i^*)$, which by (29) is

$$Q_i \ge C_i''(\boldsymbol{\theta}_i^*) - \frac{C_i'(\theta_i^*)}{P_i'(\theta_i^*)} P_i''(\theta_i^*),$$

or

$$Q_i \ge \frac{P_i'(\theta_i^*) C_i''(\theta_i^*) - C_i'(\theta_i^*) P_i''(\theta_i^*)}{P_i'(\theta_i^*)}.$$

By (27) the denominator is negative, so we can conclude the proof by showing that the numerator is negative. From (29) we have

$$\lambda_i'(\boldsymbol{\theta}_i^*) = \frac{P_i'(\boldsymbol{\theta}_i^*) C_i''(\boldsymbol{\theta}_i^*) - C_i'(\boldsymbol{\theta}_i^*) P_i''(\boldsymbol{\theta}_i^*)}{W_i [P_i'(\boldsymbol{\theta}_i^*)]^2}. \tag{31}$$

By hypothesis $0 > \lambda_i'(\boldsymbol{\theta}_i^*)$, so the numerator of (31) is negative. Since $W_i > 0$, we have $Q_i > 0$, as was to be shown.

Luenberger [1973, p.235] proves that given the constraint qualifications guaranteed in Lemma 5, the first-order necessary conditions (24) are sufficient for (at least local) optimality provided $\nabla_{\theta\theta}L(\theta^*,\lambda^*,\zeta^*)$ is positive-definite on the subspace tangent to the constraints tight at θ^*. We have shown that $\nabla_{\theta\theta}L(\theta^*,\lambda^*,\zeta^*)$ is positive-definite, which concludes the proof. We note that to prove this theorem for more general functions C_i and P_i it is sufficient that C_i be convex, and P_i and λ_i be decreasing □.

Theorem 8. For i=1,...,N, if D_i is sufficiently large, then $\lambda_i(\boldsymbol{\theta}_i)$ is convex and decreasing when $\theta_i \le \theta_i^u$.

Proof. As usual we suppress i. We write $\lambda(\boldsymbol{\theta})$ in partial fraction form making differentiation easier. We then show that for D large, the coefficients of $\lambda(\boldsymbol{\theta})$ are such that $\lambda'(\boldsymbol{\theta})<0$ for $\theta \le \boldsymbol{\theta}^u$. By long division,

$$\lambda(\theta) = \frac{F\boldsymbol{\theta}}{H} + \frac{EH-FG}{H^2} + \frac{\left(\frac{BH^2-G(EH-FG)}{H^2}\right)\boldsymbol{\theta} + A}{\theta(G+H\theta)} .$$

For any X, $\dfrac{X\boldsymbol{\theta}+A}{\boldsymbol{\theta}(G+H\boldsymbol{\theta})} = \dfrac{A}{G\boldsymbol{\theta}} + \dfrac{XG-AH}{G(G+H\boldsymbol{\theta})}$,

and so

$$\lambda(\boldsymbol{\theta}) = \frac{F\theta}{H} + \frac{EH-FG}{H^2} + \frac{A}{G\theta} + \frac{\eta}{GH^2(G+H\theta)} , \qquad (32)$$

where $\eta = GBH^2 - G^2(EH-FG) - AH^3$.

By (21) and (23) we have

$$A, B, E, F, G, H > 0, \qquad (33)$$

so certainly $\lambda(\theta)>0$ whenever $\theta>0$ (the change of variables (7) is such that $\theta>0$). Differentiating (32) gives

$$\lambda'(\theta) = \left(\frac{F}{H} - \frac{A}{G\theta^2}\right) - \frac{\eta}{GH(G+H\theta)^2}$$

and

$$\lambda''(\theta) = \frac{2A}{G\theta^3} + \frac{2\eta}{G(G+H\theta)^3} .$$

We first show that $\lambda''(\theta) > 0$ for D sufficiently large by showing $\eta > 0$ for D sufficiently large (since (33) implies all other quantities are positive). We use the following notation.

$$f(x) \sim g(x) \text{ if } \lim_{x\to\infty} \frac{f(x)}{g(x)} = 1.$$

To examine the behavior of η for large D, we examine the behavior of A, B, E, F, G and H for large D. The following are easily inferred from their definitions:

$$\left.\begin{aligned} &\lim_{D\to\infty} \delta = \infty \\ &\lim_{D\to\infty} \rho = 0 \\ &\lim_{D\to\infty} D\rho = \mu. \end{aligned}\right\} \qquad (34)$$

They imply

$$\left.\begin{aligned}
&A \sim Dh\\
&B \sim Dh\delta\\
&E \sim 2Dh\delta\\
&F \sim Dh\delta\\
&G \sim W\delta\\
&H \sim W(\mu\gamma+1)\delta\ ,
\end{aligned}\right\}\tag{35}$$

and so $\eta \sim (W\delta)(Dh\delta)W^2(\mu\gamma+1)^2\delta^2 - W^2\delta^2[(2Dh\delta)(W(\mu\gamma+1)\delta) - (Dh\delta)(W\delta)]$
$- (Dh)(W^3(\mu\gamma+1)^3\delta^3)$.

The limiting behavior of η is determined by its highest order term, which is $D\delta^4$, so

$$\begin{aligned}\eta &\sim [W^3h(\mu\gamma+1)^2 - 2W^3h(\mu\gamma+1) + W^3h]\, D\delta^4\\ &= W^3h(\mu\gamma)^2 D\delta^4.\end{aligned}$$

Since $W^3h(\mu\gamma)^2 > 0$, (34) implies $\lim\limits_{D\to\infty} W^3h(\mu\gamma)^2D\delta^4 = \infty$, so $\lim\limits_{D\to\infty} \eta = \infty$ also. Thus, for D sufficiently large, $\eta > 0$, and so $\lambda''(\theta) > 0$, as was to be shown. Hence $\lambda(\theta)$ is convex. To show that $\lambda'(\theta) < 0$ for $\theta \leq \theta^u$, it is sufficient to show $\lambda'(\theta^u) < 0$. We have

$$\lambda'(\theta^u) = \xi - \frac{\eta}{GH(G+H\theta^u)^2}\ ,$$

where
$$\xi = \frac{F}{H} - \frac{A}{G(\theta^u)^2}\ .$$

Since for $\eta>0$ large D, we need only show that $\xi<0$ for large D to conclude $\lambda'(\theta) < 0$ for $\theta \leq \theta^u$. Relations (35) imply that

$$\xi \sim \frac{Dh\delta}{W(\mu\gamma+1)\delta} - \frac{Dh}{W\delta\cdot\dfrac{e^{2\gamma\mu^*}}{\delta^2}}$$

$$= \frac{Dh}{W}\left(\frac{1}{\mu\gamma+1} - \frac{\delta}{e^{2\gamma\mu^*}}\right).$$

Relations (34) imply the quantity in parentheses tends to $-\infty$ as $D\to\infty$, and so

$$\lim_{D\to\infty} \xi = \lim_{D\to\infty} \frac{Dh}{W}\left(\frac{1}{\mu\gamma+1} - \frac{\delta}{e^{2\gamma\mu^*}}\right) = -\infty.$$

Thus, $\xi < 0$ for D sufficiently large, and so the proof is complete □.

Thus, Theorems 7 and 8 imply that if D_i is large enough that λ_i is decreasing, then the first-order conditions (24) are sufficient for (at least local) optimality. We exploit this in our algorithm, which is described in the next section.

3. GKD ALGORITHM

Our algorithm is based upon local duality theory of nonlinear programming (see Luenberger [1973, pp.312-320]). Suppose x* is a local minimum (at which the constraint qualification described in Lemma 5 holds) of the nonlinear program

$$\left.\begin{array}{l} \text{minimize } f(x) \\ \text{subject to } g(x)=0, \end{array}\right\} \qquad (36)$$

with associated Kuhn-Tucker constraint multiplier λ^*. Writing the Lagrangian $L(x,\lambda) = f(x)-\lambda g(x)$, the NLP dual to (36) is defined as

$$\underset{\lambda}{\text{maximize}}\ \psi(\lambda),$$

where (37)

$$\psi(\lambda) = \underset{x \text{ near } x^*}{\text{minimum}}\ L\ (x,\lambda).$$

(We mention that (36) and (37) can be written, respectively, as $\min_x \max_\lambda L(x,\lambda)$ and $\max_\lambda \min_x L(x,\lambda)$, which illustrates their dual nature).

If $L(x,\lambda^*)$ is twice continuously-differentiable and convex at x^* (and hence convex in a neighborhood of x^*), then the following can be shown to be true:

(1) $\psi(\lambda)$ is twice continuously-differentiable and convex at λ^*.

(2) λ^* uniquely solves (37).

(3) If we denote by $x(\lambda)$ the (unique) value of x minimizing the right-hand side of $\psi(\lambda)$ in (37), then $x^*=x(\lambda^*)$.

Thus rather than solving (36) one could solve (37), and do this by solving

$$\nabla\psi(\lambda)=0,$$

which can be shown to be equivalent to solving

$$g(x(\lambda))=0. \tag{38}$$

Generally this approach is not useful computationally because the function $x(\lambda)$ is often difficult to compute. This method may be quite efficient, however, if $x(\lambda)$ is reasonably easy to compute and the dimension of λ is smaller than that of x. One problem with such structure is the "Generalized Knapsack Problem," which is

$$\left.\begin{aligned} &\text{minimize } f(x) = \sum_{i=1}^{N} f_i(x_i) \\ &\text{subject to } g(x) = \sum_{i=1}^{N} g_i(x_i) = 0. \end{aligned}\right\} \tag{39}$$

Clearly (17) is a Generalized Knapsack Problem with upper-bounded variables. These bounds do not change the essential structure of the problem or its solution by solving its dual, and the details are left to Theorem 9.

For (39), $x(\lambda)$ is found by solving the system of equations

$$\nabla_x L(x,\lambda)=0,$$

where

$$\lambda_i^L = \lambda_i(\boldsymbol{\theta}_i^u).$$

For convenience, we assume that the items are reordered, if necessary, so that

$$\lambda_1^L \geq \lambda_2^L \geq \ldots > \lambda_N^L. \tag{41}$$

Theorem 9. If either $P(\theta^u) \geq 0$ or all D_i are sufficiently large so that the hypotheses of Theorem 8 hold, then the following algorithm is well-defined and converges to a (possibly local) minimum of (17).

GKD Algorithm.

(1) If $P(\boldsymbol{\theta}^u) \geq 0$ set i=N, and go to step (6). Otherwise set i=0.

(2) Set $\lambda^L = \lambda_{i+1}^L$ $(= \max\left\{\lambda_{i+1}^L,\ldots,\lambda_N^L\right\}$, by (41)).

(3) Define the function $\hat{P}_i$ on $[\lambda^L, +\infty)$ by

$$\hat{P}_i(\lambda) = P(\boldsymbol{\theta}_1^u,\ldots,\boldsymbol{\theta}_i^u, t_{i+1}(\lambda),\ldots,t_N(\lambda)).$$

(4) If $\hat{P}_i(\lambda^L) \leq 0$, go to step (5).

Otherwise set i=i+1, go to step (2).

(5) Solve $\hat{P}_i(\lambda) = 0$ for $\lambda^* \varepsilon [\lambda^L, \infty)$.

(6) Set $\boldsymbol{\theta}_j^* = \begin{cases} \boldsymbol{\theta}_j^u & j=1,\ldots,i \\ t_j(\lambda^*) & j=i+1,\ldots,N, \end{cases}$

$$S_j^* = \boldsymbol{\mu}_j^* - \frac{1}{\boldsymbol{\gamma}_j} \log \boldsymbol{\theta}_j^*$$

$$s_j^* = S_j^* - D_j.$$

Proof. Steps (1) and (2) are certainly well-defined. By definition of λ^L, $\lambda \geq \lambda^L$ is in the domain of $t_{i+1}(\lambda),\ldots, t_N(\lambda)$, so these are all well-defined. Hence so is $P(\lambda)$, and so step (3) us well-defined. Step (4) is well-defined, except for the possibility that i may become larger than N. But if i=N, then we have $P(\theta_1^u,\ldots,\theta_N^u) > 0$, which is equivalent to $P(\boldsymbol{\theta}^u) > 0$. But step (1) guarantees $P(\boldsymbol{\theta}^u) < 0$, so i will never exceed N.

Step (5) requires justification. We show that $\hat{P}_i'(\lambda)>0$ on $[\lambda^L,+\infty)$ and $\hat{P}_i(+\infty)>0$. These with step (4) imply that there is a unique solution in $[\lambda^L,+\infty)$ to $\hat{P}_i(\lambda)=0$. We have

$$\hat{P}_i(\lambda) = P(\theta_1{}^u,\ldots,\theta_i{}^u,\ t_{i+1}(\lambda),\ldots,t_N\ (\lambda)),$$

so

$$\hat{P}_i'(\lambda) = \sum_{j=i+1}^{N} W_j P_j'(\theta_j)\ t_j'(\lambda),\ \text{where}\ \theta_j = t_j(\lambda),$$

by the chain rule. Relation (27) implies that $\hat{P}_j'(\theta_j)<0$, and clearly $t_j'(\lambda)<0$, so indeed $\hat{P}_i'(\lambda)>0$. Stage i is the first time that $P_i(\lambda^L)\leq 0$. If i=0, then

$$\begin{aligned}\hat{P}_i(\infty) &= P(t_1(\infty),\ldots,t_N(\infty))\\ &= P(0,\ldots,\ 0) > 0,\end{aligned}$$

by (26). And if $i \geq 1$, then $\hat{P}_{i-1}(\lambda^L) > 0$, so

$$\begin{aligned}0 < \hat{P}_{i-1}(\lambda^L) &= P(\theta_1{}^u,\ldots,\boldsymbol{\theta}_{i-1}^u,t_i,\ldots,t_N)\\ &= P(\theta_1{}^u,\ldots,\theta_i{}^u,t_{i+1},\ldots,t_N),\end{aligned}$$

since at stage i-1, $\lambda^L=\lambda_i{}^L$, and $t(\lambda_i{}^L) = \theta_i{}^u$.

Hence, $0 < P(\boldsymbol{\theta}_1{}^u,\ldots,\boldsymbol{\theta}_i{}^u,t_{i+1},\ldots,t_N)$.

We have

$$\begin{aligned}\hat{P}_i(\infty) &= P(\theta_1{}^u,\ldots,\theta_i{}^u,t_{i+1}(\infty),\ldots,t_N(\infty))\\ &= P(\theta_1{}^u,\ldots,\theta_i{}^u,0,\ldots,0).\end{aligned}$$

By (27) P is decreasing in each argument, so $\hat{P}_i(\infty) > \hat{P}_{i-1}(\lambda^L) > 0$, as was to be shown.

Step (6) and (7) are well-defined, so the entire algorithm is verified. If $P(\theta^u)\geq 0$ at step (1), then by Theorem 4, $\theta^*=\theta^u$ is globally optimal for (17). If $P(\theta^u)<0$ at step (1), the fact that the functions λ_j and t_j are inverses with the fact that $\lambda_j{}^L=\lambda_j(\theta_j{}^u)$ imply that the algorithm terminates with $0<\theta^*\leq\theta^u$ and $\lambda^*\geq 0$ satisfying

(a) $\lambda_j(\theta_j^*) = \lambda_j^L$, $j=1,\ldots,i$

$\lambda_j(\theta_j^*) = \lambda^*$, $j=i+1,\ldots,N$

(b) $P(\theta^*) = 0$.

But (41) implies $\lambda_j(\boldsymbol{\theta}_j^*) \geq \lambda^L$ for $j=1,\ldots,i$, and certainly $\boldsymbol{\lambda}^L \geq \lambda^*$. Moreover, since $\theta_j^*=\boldsymbol{\theta}_j^u$, for $j=1,\ldots,N$, θ^* satisfies (24). By Theorems 6, 7 and 8, these equations imply the local optimality of θ^* for (17).

Recall that we reformulated the multi-item inventory problem using the one-to-one change of variables given by (7). Our GKD Algorithm finds an optimal $\boldsymbol{\theta}^*$ for (17), and this gives rise to an optimal S^* via the inverse of the change of variables (17), which is

$$S_i = \mu_i^* - \frac{\log\boldsymbol{\theta}_i}{\gamma_i},$$

as given in step (6) □.

The algorithm is computationally efficient in that it only requires solving $\hat{P}_i(\lambda)=0$, where $\hat{\mathbf{P}}_i(\lambda)$ is a real-valued function of one real variable. The computation of the functions $t_j(\lambda)$ must, however, be done numerically (the details are in the Appendix to this chapter). Moreover, the hypotheses of Theorem 8 are easy to verify and guarantee that the algorithm is well-defined and converges to a minimum of (17). The proof of Theorem 8 shows that is sufficient to check that $\lambda_i'(\boldsymbol{\theta}_i^u) < 0$ for each i. The extension of the algorithm when $\lambda_i'(\boldsymbol{\theta}_i^u) \geq 0$ is discussed in Section 5.

4. POLICY PERFORMANCE

In order to illustrate the performance of the (s,S) policies generated by our algorithm, we performed the following test. We examined fifteen 2-item inventory systems with the parameter settings

$$K_1 = K_2 = 32$$

$$k_1 = k_2 = 0$$

$$h_1 = .05, \ \frac{h_2}{h_1} = 1, 10, 25, 50, 100$$

$$\mu_1 = 3, \ \frac{\mu_2}{\mu_1} = \frac{1}{3}, 1, 3$$

$$\frac{\sigma_1^2}{\mu_1} = \frac{\sigma_2^2}{\mu_2} = 9$$

$$W_1 = W_2 = \frac{1}{2} \ .$$

We chose these parameter settings for three reasons

1. Most of them give rise to large D (much larger than 1.5μ). Since our approximations and theorems generally hold for large D, it should perform well on these systems.
2. These parameters are realistic, in that they reflect costs associated with many actual inventory systems.
3. As in Mitchell [1982], our algorithm should perform effectively when there are both "expensive" and "inexpensive" items in the system. This holds for those systems we considered with $\frac{h_2}{h_1}$ large.

The comparison we used assumes the underlying distribution is Negative Binomial (the discrete version of a Gamma distribution, which is skewed for our parameter settings), specifies a service-level α, and uses the Veinott-Wagner exact algorithm (Veinott and Wagner [1965]) to compute optimal policies for both items in each system using the same service-level α. We ran our algorithm using α. Then we compared the holding plus set-up costs generated by the Veinott-Wagner algorithm and our algorithm. The results are displayed in Table 1.

We make the following observations.

1. The service-level α is not the same for all runs, but varies from .89 to .97. The average service-level for the fifteen runs is .93. There are two reasons for this. First, for some of these items, the policy with the smallest service-level, (0,D), has a very high service-level, so that the system-wide service-level is high. For other items, the smallest service-level is small. It turns out that to specify a uniform service-level is to specify α=.97 (since one of the systems has this as the smallest service-level for both items), which is unreasonably large. The second reason is that the Logistic approximation we use in our algorithm does not always generate policies with service-level α when the actual distribution is Negative Binomial. However, preliminary tests on larger systems (32 items) seem to indicate that this problem is less severe for larger systems.

2. Our approximation for D_i is almost exact for every single item. This suggests that the asymptotic approximations we are using should be fairly accurate. For all items D_i is sufficiently large to guarantee that the hypotheses of Theorem 7 are satisfied, so all policies are (possibly local) minima (not saddle points).

3. Over half the systems had an item with $s < 0$ for the policy generated by the Veinott-Wagner algorithm. Our algorithm can only consider policies with $s \geq 0$, but its policies still compared very favorably against those generated by the Veinott-Wagner algorithm.

4. As expected, there was a cost reduction for all systems, except the ones with identical holding costs. Moreover, the greatest cost decreases were for those systems with $\frac{h_2}{h_1}$ large.

This initial investigation demonstrates that inventory-operating costs can be reduced significantly when a system-wide service level is specified, rather than specifying an identical service-level for each item.

$\frac{h_2}{h_1}$ \ $\frac{\mu_2}{\mu_1}$	$\frac{1}{3}$	1	3
100	40	33	31
50	26	23	25
25	9	14	19
10	1	7	10
1	0	0	0

Table 1

Percent decrease in holding + set-up costs

Assumptions:

$K_1=K_2=32$, $k_1=k_2=0$, $h_1=.05$

$\frac{\sigma_1^2}{\mu_1} = \frac{\sigma_2^2}{\mu_2} = 9$, $W_1=W_2=\frac{1}{2}$, $\mu_1=3$

all demands Negative Binomial.

5. GKD ALGORITHM MODIFICATION WHEN D_i IS SMALL

Theorem 8 of Section 2 contains the most general condition under which we were able to prove (see Theorem 7) that the GKD Algorithm converges to a (possibly local) minimum of (17). This condition is that for all items i=1,...,N the functions $\lambda_i(\theta_i)$ are decreasing on $(0, \theta_i^u)$. We give our recommendations when this condition may not hold for some items in the inventory system.

Recall from the proof of Theorem 8 that $\lambda_i(\theta_i)$ is convex on $(0, \theta_i^u)$ when $\eta_i>0$ and is decreasing on $(0, \theta_i^u)$ when both $\eta_i>0$ and $\xi_i<0$. Moreover, $\lim_{D_i\to\infty} \eta_i=\infty$ and $\lim_{D_i\to\infty} \xi_i=-\infty$, so $\lambda_i(\theta_i)$ is decreasing on $(0, \theta_i^u)$ when D_i is sufficiently large. The GKD Algorithm (given in Theorem 9) exploits this property by assuming the existence of the inverse function of λ_i, which is denoted by t_i.

The question is what to do when D_i is not large enough to guarantee $\eta_i>0$ and $\xi_i<0$. In either case it is impossible to guarantee that λ_i is decreasing, and hence invertible, so step (3) of the GKD Algorithm may not be well-defined. The algorithm may converge, but there is no guarantee that it will converge to a local minimum.

We recommend the following modifications when $\eta_i<0$ or $\xi_i>0$. If $\xi_i>0$ but $\eta_i>0$, then λ_i is still convex but not decreasing on $(0, \theta_i^u)$. It is always true (and easy to verify) that $\lambda'(0)=-\infty$, so this case is illustrated in Figure 1.

In this case we recommend that the domain of $\lambda_i(\theta_i)$ be shortened to $(0, \hat{\theta}_i^u)$, where $\hat{\theta}_i^u$ is the largest value of θ_i for which $\lambda_i(\theta_i)$ is decreasing. This has the effect of introducing the additional constraint $S_i\ge S_i^L$, where $S_i^L\ge D_i$. Since the S_i are lower bounded anyway by D_i, this constraint does not change the structure of the problem at all. In particular, the GKD Algorithm will still converge to a (possibly local) minimum.

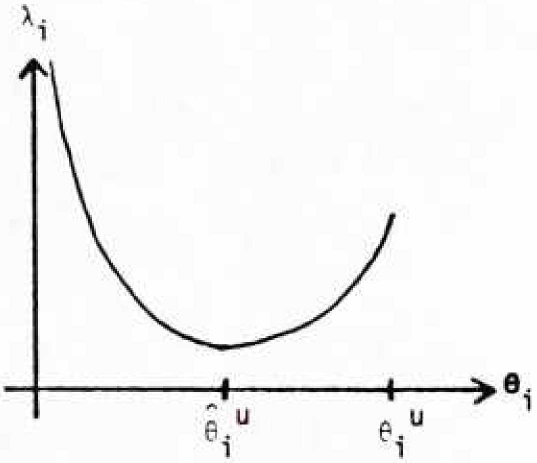


Figure 1

Computationally, $\hat{\theta}_i^u$ can be found by increasing the given lower bound D_i on S_i a unit at a time until $\hat{\theta}_i^u$ is found. This procedure is described in detail at the end of this section, and computational experience on typical 32-item inventory systems is reported in the next chapter.

Although we have in this case a recommendation that still assures that the GKD Algorithm will converge, the case when $\eta_i<0$ poses greater difficulty. The λ_i may not be convex, in which case we have been unable to find a modification of the problem that guarantees convergence to a local minimum. We essentially ignore this case and use the algorithm as it is (making the recommended modification when $\xi_i>0$). Our computational experience reported in the next chapter seems to indicate that this is not a problem in many cases. Since η_i is easy to compute prior to using the GKD Algorithm, we recommend doing so in order to be alerted to possible problems when $\eta_i<0$.

In light of all this, the only recommended change involves the computation of θ_i^u. The GKD Algorithm is just as given in Theorem 9, with the following step inserted between steps (1) and (2):

```
(1.1).  For $j=1,\ldots,N$:
        Set $S_j^L=D_j$
        (1.1.1).  Set $\theta_j^u = e^{-\gamma_j(S_j^L-\mu_j*)}$.
                  If $\lambda_j'(\theta_j^u)<0$, go to step (1.1.2).
                  Otherwise, set $S_j^L=S_j^L+1$.
                  Go to step (1.1.1).
        (1.1.2).  Next $j$.
```

6. CONCLUSIONS

We have formulated a constrained NLP to solve an (s,S) inventory model with a service-level constraint rather than shortage costs. This model has the advantage that a manager can more easily specify service objectives than shortage costs. We derived necessary and sufficient conditions for an optimal (s,S) policy. Using asymptotic approximations (where D=S-s is large) and assuming Logistically distributed demands, we derived a computationally efficient algorithm to find approximately optimal (s,S) policies. We showed that for some inventory systems there is a substantial cost savings when using this algorithm instead of the Independent and Identical Service approach that is often used in practice. We also gave recommendations to modify our algorithm when D is not sufficiently large to guarantee its convergence.

NOTATION

	Page
K_i	18
c_i	18
k_i	18
h_i	18
Φ_i	19
ϕ_i	19
μ_i	19
σ_i	19
N	19
W_i	19
α	19
(s,S)	19
D_i	20
Φ_i^*	20
ϕ_i^*	20
μ_i^*	20
σ_i^*	20
$C_i(S_i,D_i)$	21
$P_i(S_i,D_i)$	21
M_i	21
m_i	21
$C(S,D)$	22
$P(S,D)$	22

	Page
ρ_i	22
θ_i	25
γ_i	25
$C_i(\theta_i)$	25
$P_i(\theta_i)$	25
δ_i	26
θ_i^u	29
$C(\theta)$	30
$P(\theta)$	30
A_i	33
B_i	33
E_i	33
F_i	33
G_i	33
H_i	33
$\lambda_i(\theta_i)$	33
$L(\theta^*,\lambda^*,\zeta^*)$	34
$t_i(\lambda_i)$	42
λ_i^L	42
λ^L	43
$\hat{P}_i(\lambda)$	43

its unit holding cost h). This system is described in Figure 2, where the items are ordered in decreasing order by hμ. Essentially the first seven items are high-value items (have a very high annual dollar demand), and the other twenty-five items are low-valued items.

As a base-case, we specified inventory parameters as follows. For i=1,...,N,

$$K_i = 24$$
$$W_i = \frac{1}{32}$$
$$k_i = 4$$
$$\frac{\sigma_i{}^2}{\mu_i} = 9$$

and

$$\mu_1, \mu_5, \mu_9, \ldots, \quad \mu_{29} = 16$$
$$\mu_2, \mu_6, \mu_{10}, \ldots, \mu_{30} = 8$$
$$\mu_3, \mu_7, \mu_{11}, \ldots, \mu_{31} = 4$$
$$\mu_4, \mu_8, \mu_{12}, \ldots, \mu_{32} = 2 .$$

Of course, h_i is determined by μ_i from the value $(h\mu)_i = h_i \mu_i$.

We are interested primarily in the algorithm performance at a service-level of 85%, one managers often prefer. Our criterion of comparison is to assume that the demand distribution is Negative Bincmial and compare the total expected cost associated with the (s,S) policies that the GKD Algorithm produces with the total expected cost of the policies given by optimally setting all the (s,S) policies so that every item has a service-level of 85%. We call this the Identical Service Approach, and it is an approach often used by managers. We intended to use the Veinott-Wagner [1965] Algorithm (also Kaufman

[1976]) to compute the optimal policies, but the amount of computation required was impractical because the optimal policies often had S_i too large.

Instead we used the Power Approximation of Ehrhardt [1976] and [1979] to compute approximately optimal policies. We chose this approximation because extensive numerical investigations in Ehrhardt [1976], and Klincewicz [1976a] and [1976b] showed this to be both the most accurate and most easily computed approximately optimal policy available. We use the version of the Power Approximation given in Mosier [1981] because of the technical improvements made. The Power Approximation requires the specification of shortage costs rather than the service-level. Letting α denote the service-level specified for the inventory system (using the Identical Service Approach), we specified the shortage costs p_i using the expression (see Ehrhardt [1977, p.8])

$$\alpha = \frac{p_i + 0.0695h_i}{p_i+h_i} .$$

This expression is an empirical improvement over the well-known relationship (see Veinott and Wagner [1965])

$$\alpha = \frac{p_i}{p_i+h_i} .$$

Because both the GKD Algorithm and the Power Approximation are approximations, it was impossible to realize the exact service-level specified. We therefore specified various service-levels, and interpolated intermediate service-levels. Since we used the Power Approximation in the GKD Algorithm to compute the values for D, we only compared the expected holding costs for this algorithm and the Power Approximation, the expected replenishment costs being identical (Typically the expected replenishment cost accounts for about half of the total expected cost). The result of this experiment is displayed in Figure 3. There is a significant decrease in expected holding costs at and near a service-level of 85%.

In order to investigate the sensitivity of expected holding cost to the various parameters, we performed four additional experiments in which we modified one parameter (or parameter group) at a time in our base-case model. The same experimental design was used for these experiments as for the base-case.

Figure 4 describes the result when the means were changed to

$$\mu_1, \mu_3, \ldots, \mu_{31} = 16$$

$$\mu_2, \mu_4, \ldots, \mu_{32} = 8.$$

As can be seen, there was a very slight degradation in the GKD Algorithm performance. The cost savings are still substantial.

Figure 5 describes the result when the base-case was altered so that

$$k_i = 8, \ i = 1, \ldots, 32.$$

Again there was only a slight degradation in cost.

Figure 6 is the base-case with

$$\frac{\sigma_i^2}{\mu_i} = 3, \ i = 1, \ldots, 32.$$

The cost degradation is greater than in previous experiments, but the cost savings remains substantial.

Figure 7 describes the change

$$\mu_1 \cdot \mu_2, \ldots, \mu_{16} = 16$$
$$\mu_{17}, \mu_{18}, \ldots, \mu_{32} = 8.$$

The cost savings are still substantial, but not as great as in previous experiments.

As a final experiment, we did a worst-case type experiment, in which all of the following changes were made:

$$k_i = 8$$
$$\frac{\sigma_i^2}{\mu_i} = 3$$
$$\mu_1, \ldots, \mu_{16} = 16$$
$$\mu_{17}, \ldots, \mu_{32} = 8.$$

This experiment is described in Figure 8. The degradation is substantial, but the cost savings are still large. The behavior of the GKD Algorithm policies for service-levels larger than about 87%, where the total cost is greater, seem to be caused by the error in approximating the Negative Binomial demand by a Logistic distribution (Section 2 of the previous chapter). We address this issue at the end of this chapter.

We make the following general observations concerning the performance of the GKD Algorithm for these 32-item inventory systems.

1. The total holding cost savings are substantial at and near a service level of 85%. The algorithm is an improvement over the Identical Service Approach often used by managers, and significantly reduces the overall

cost of maintaining an inventory system subject to a service-level constraint.

2. In most cases the policies produced by the GKD Algorithm degrade sharply around a service-level of 86% or 87%, as is evidenced by the sudden sharp increase of the holding cost. This suggests that the GKD Algorithm may not produce substantial cost savings when a high overall service-level is specified.

3. The GKD Algorithm seems to perform best when there is substantial variation in demands means, when the leadtime is not too large, and when the variance to mean is not too small. The degradation in performance is due to the error in approximating a Negative Binomial demand distribution by a Logistic distribution. This is especially a problem when there are certain patterns in the demand means and large leadtimes. This issue is discussed in detail in the next chapter.

3. SINGLE-ITEM SERVICE-LEVELS

For the six experiments described above, we also examined the single-item service-levels of the policies produced by the GKD Algorithm. For each inventory system we chose that experiment that had the average service-level closest to 85% and plotted the items (in decreasing order of their annual dollar demand, $h\mu$) against their optimal service-level. The results are displayed in Figures 9, 10, 11, 12, 13 and 14. We make the following observations concerning them.

1. Typically the items in the inventory system are skewed, in

that the seven high-value items (i=1,...,7) have a service-level considerably below 85%, and most of the twenty-five low-value items have a service-level considerably above 85%. In some cases there are a few items with service-levels near 85%.

2. There is an obvious negative correlation between the value $h_i\mu_i$ and the service-level for item i. This indicates that low-value items should be stocked high and high-value items should be stocked much lower, rather than stocking all items the same.

3. There is an obvious cyclical pattern corresponding to the cyclical pattern of the means. For the high-value items, the service-levels are higher for those items with smaller means, and for the low-value items, the service-levels are lower for those items with the smaller means. This suggests that it may be possible to well-approximate the ordering of the items by their service-levels using only $h_i\mu_i$ and μ_i.

4. ITEMS WITH D_i TOO SMALL

The discussion in Section 5 of the previous chapter indicates that a sufficient, but not necessary, condition that the GKD Algorithm converge to a (possibly local) minimum (17) is that $\xi_i<0$ and $\eta_i>0$. These are guaranteed if D_i is sufficiently large. If D_i is not large enough to guarantee that $\xi_i<0$, it is sufficient to in effect add the additional constraint $S_i \geq S_i^L$ in order to still guarantee that the algorithm will converge to a local minimum. If $\eta_i>0$, no recommendation could be given to guarantee algorithm convergence to a local minimum. In this chapter we report on those items in the six inventory systems examined that violate either $\xi_i<0$ or $\eta_i>0$.

Interestingly enough, there is not a single item in any inventory system with $\eta_i<0$, and so the GKD Algorithm converged to a (possibly local) minimum in every experiment. Each system contained items with $\xi_i>0$, however, necessitating adding constraints $S_i \geq S_i^L$, as discussed in the previous chapter. This is reported in Table 2.

5. CONCLUSIONS

We have found that for inventory systems typical of ones found in the real-world, there is a significant cost savings when the GKD Algorithm is used rather than the Identical Service Approach often used by managers. There are two issues that still need to be addressed.

First, the behavior observed in Figure 8 (the identical service approach performed better than the GKD Algorithm) needs to be better understood. This is especially necessary in order that practical recommendations may be given as to what types of inventory systems one can expect substantial cost reductions when using the GKD Algorithm rather than the identical-service approach. Moreover, insights into the behavior observed in Figure 8 may suggest modifications to the GKD Algorithm that would enable it to perform better in situations like those which prevail in Figure 8. This issue is addressed in the next chapter.

Second, methods of large-scale inventory aggregation need to be explored so that the GKD Algorithm can be used in an inventory system with perhaps many thousands of items. Only then can the algorithm be of practical value. This issue is also addressed in the next chapter.

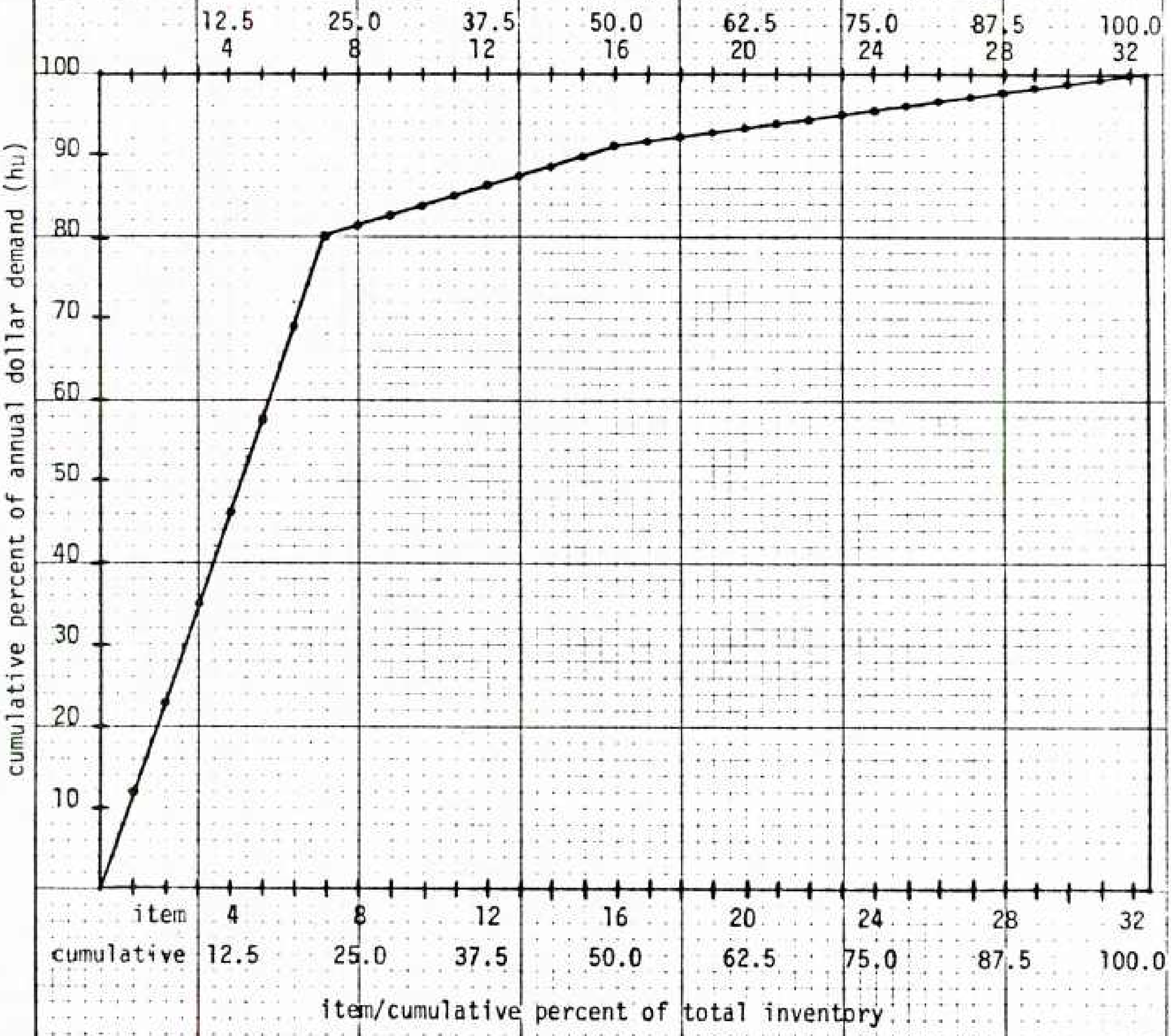
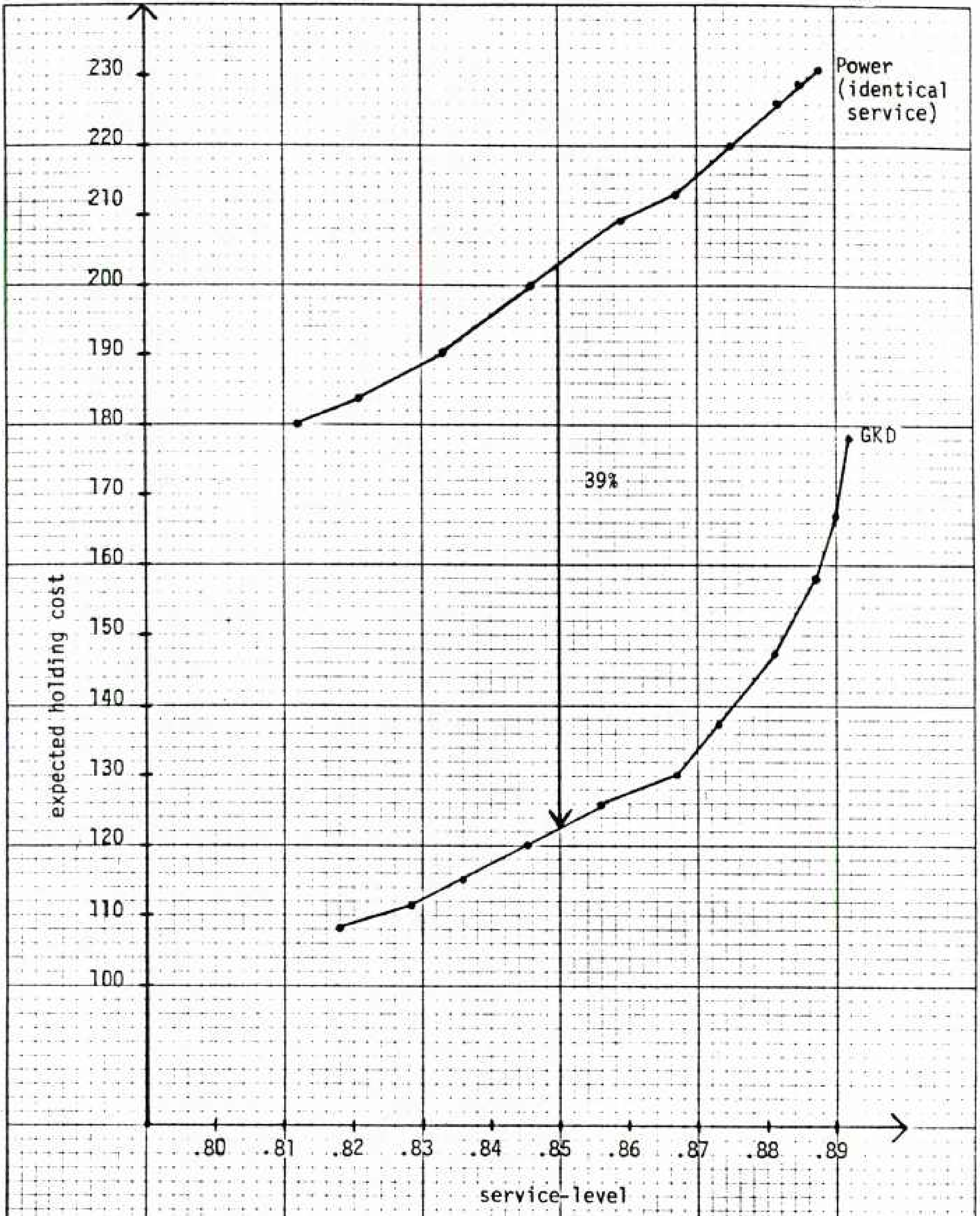

12.5
4
25.0
8
37.5
12
50.0
16
62.5
20
75.0
24
87.5
28
100.0
32
100
90
80
70
60
50
40
30
20
10
cumulative percent of annual dollar demand (hu)
item
4
8
12
16
20
24
28
32
cumulative
12.5
25.0
37.5
50.0
62.5
75.0
87.5
100.0
item/cumulative percent of total inventory

Figure 2

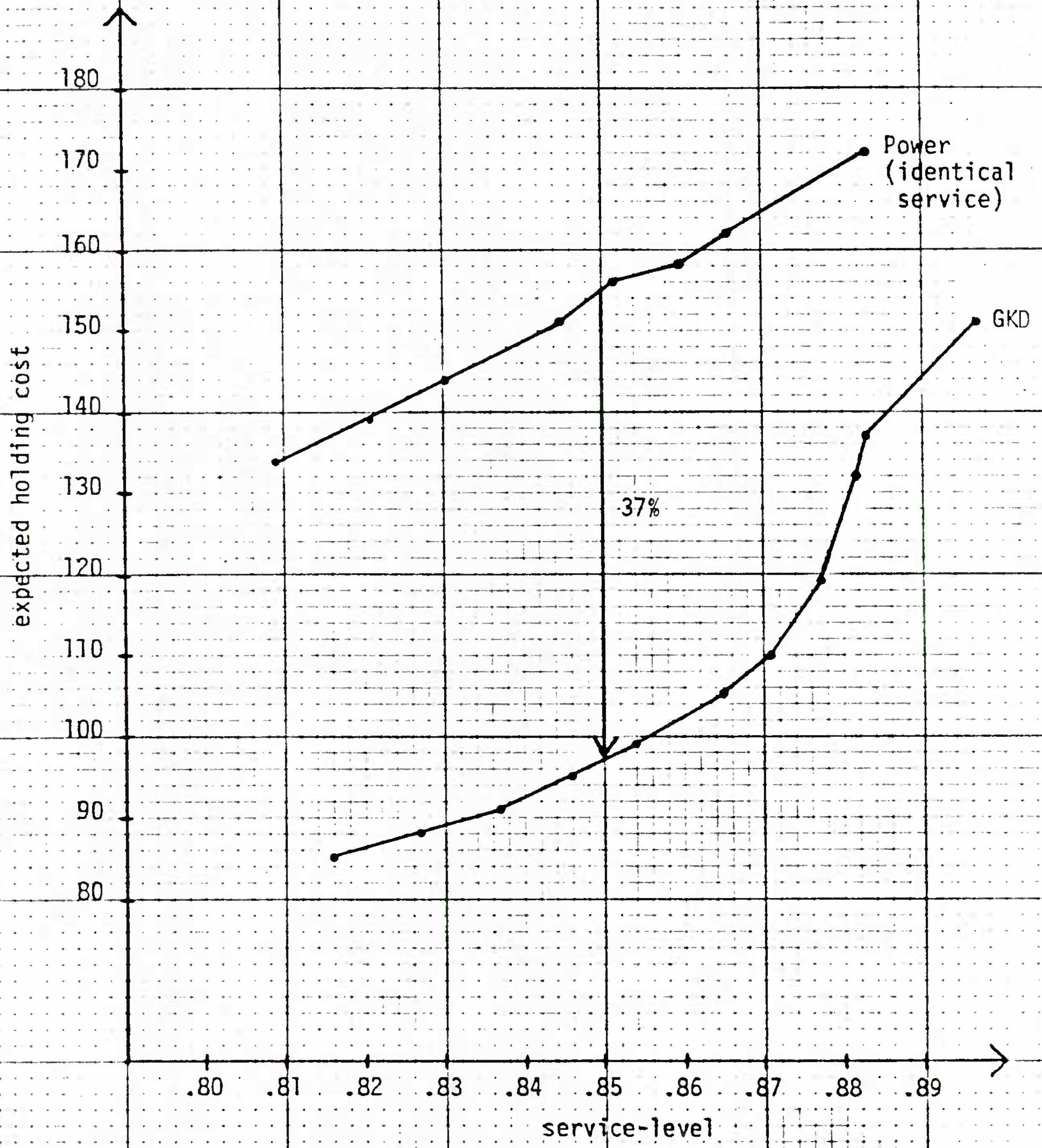


32 items, $K_i=24$, $W_i=\frac{1}{32}$, $k_i=4$,

$\frac{\sigma_1^2}{\mu_i} = 9$, μ_i: 16, 8, 4, 2, . . . ,16, 8, 4, 2

D_i using Power Method

Figure 3

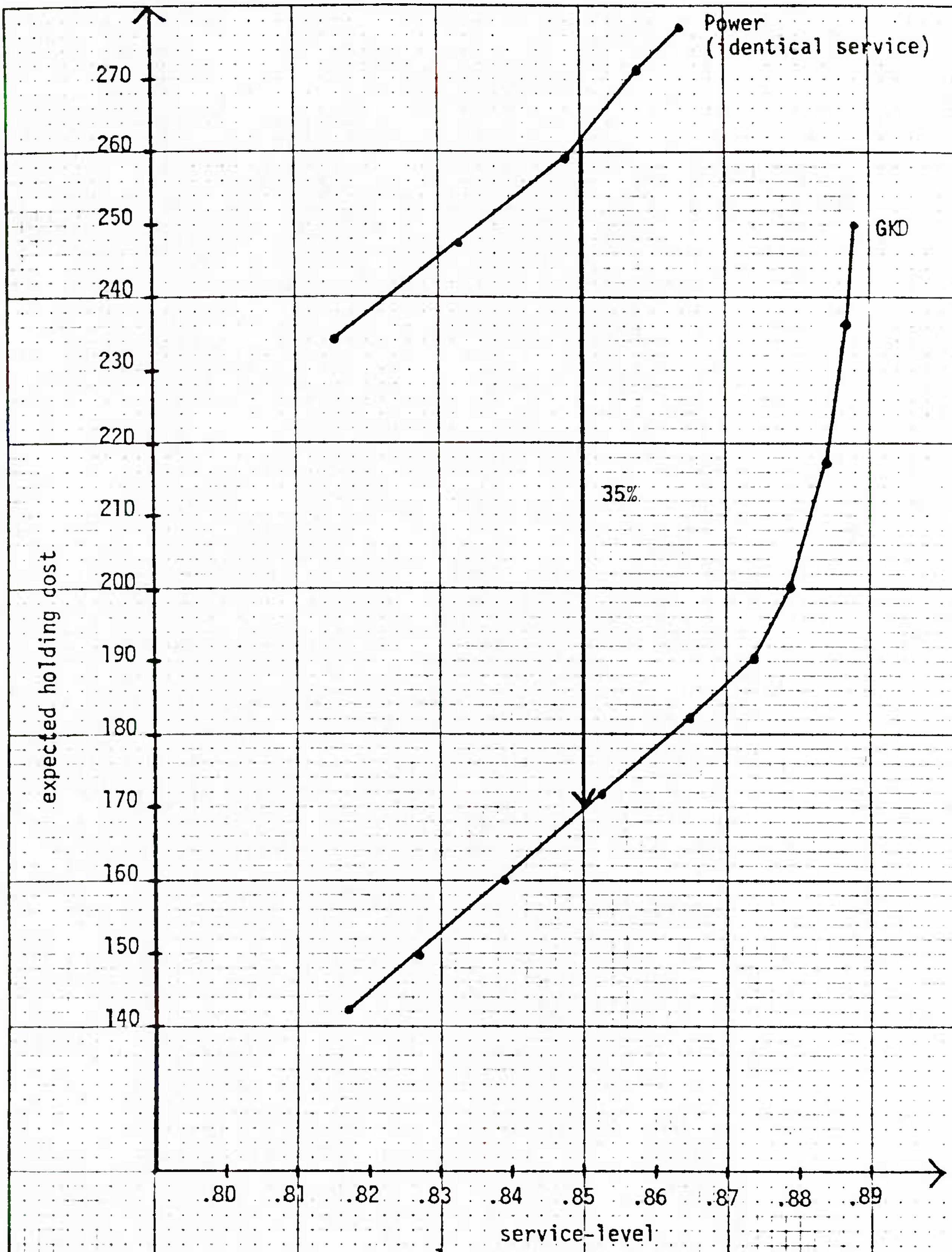


32 items, $K_i=24$, $W_i = \frac{1}{32}$, $k_1=4$

$\frac{\sigma_i^2}{\mu_i} = 9$, μ_i: 16, 8,....,16, 8

D_i using Power Method

Figure 4

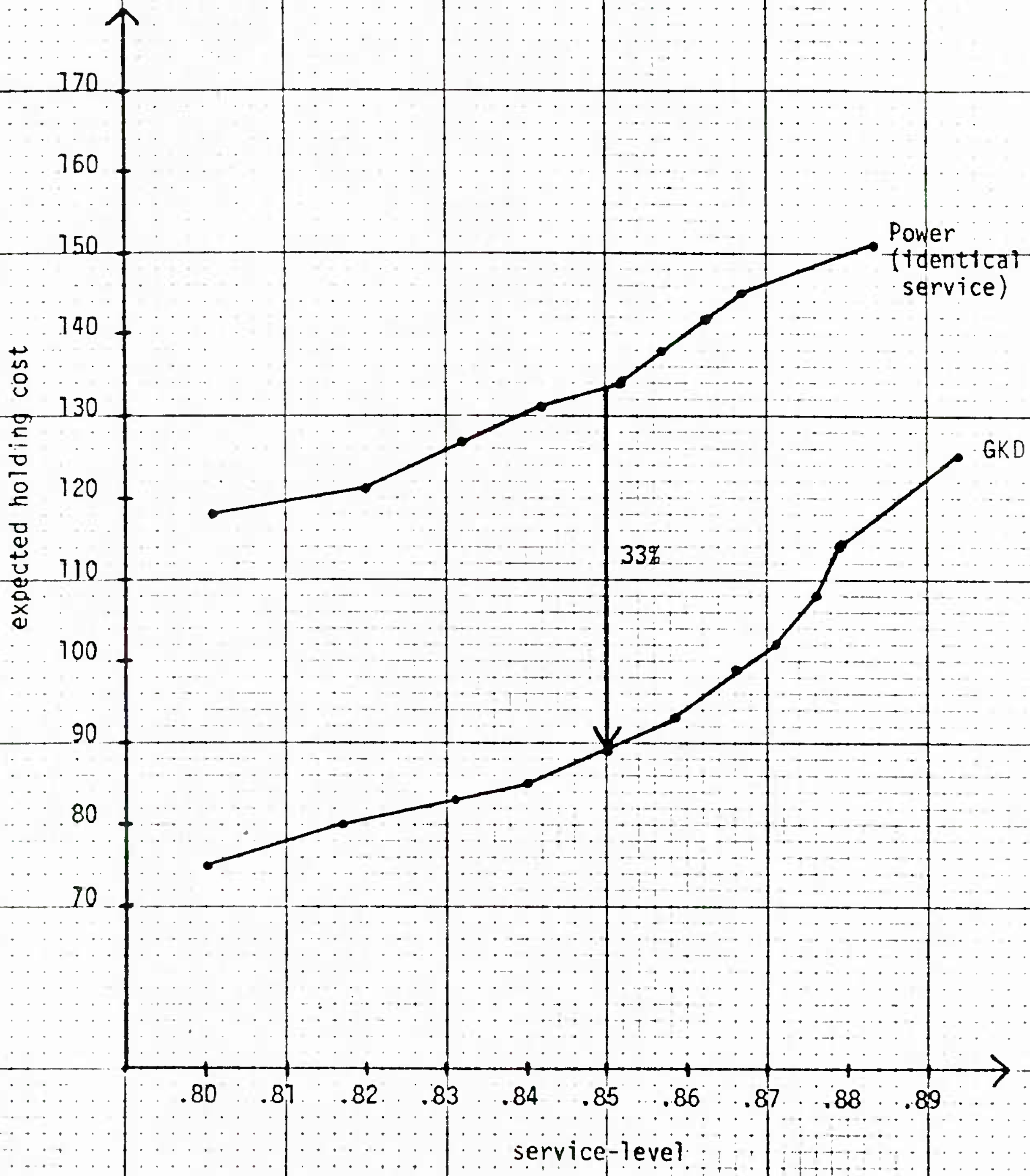


32 items, $K_i=24$, $W_i = \frac{1}{32}$, $k_i=8$

$\frac{\sigma_i^2}{\mu_i} = 9$, μ: 16, 8, 4, 2, . . . , 16, 8, 4, 2

D_i using Power Method

Figure 5

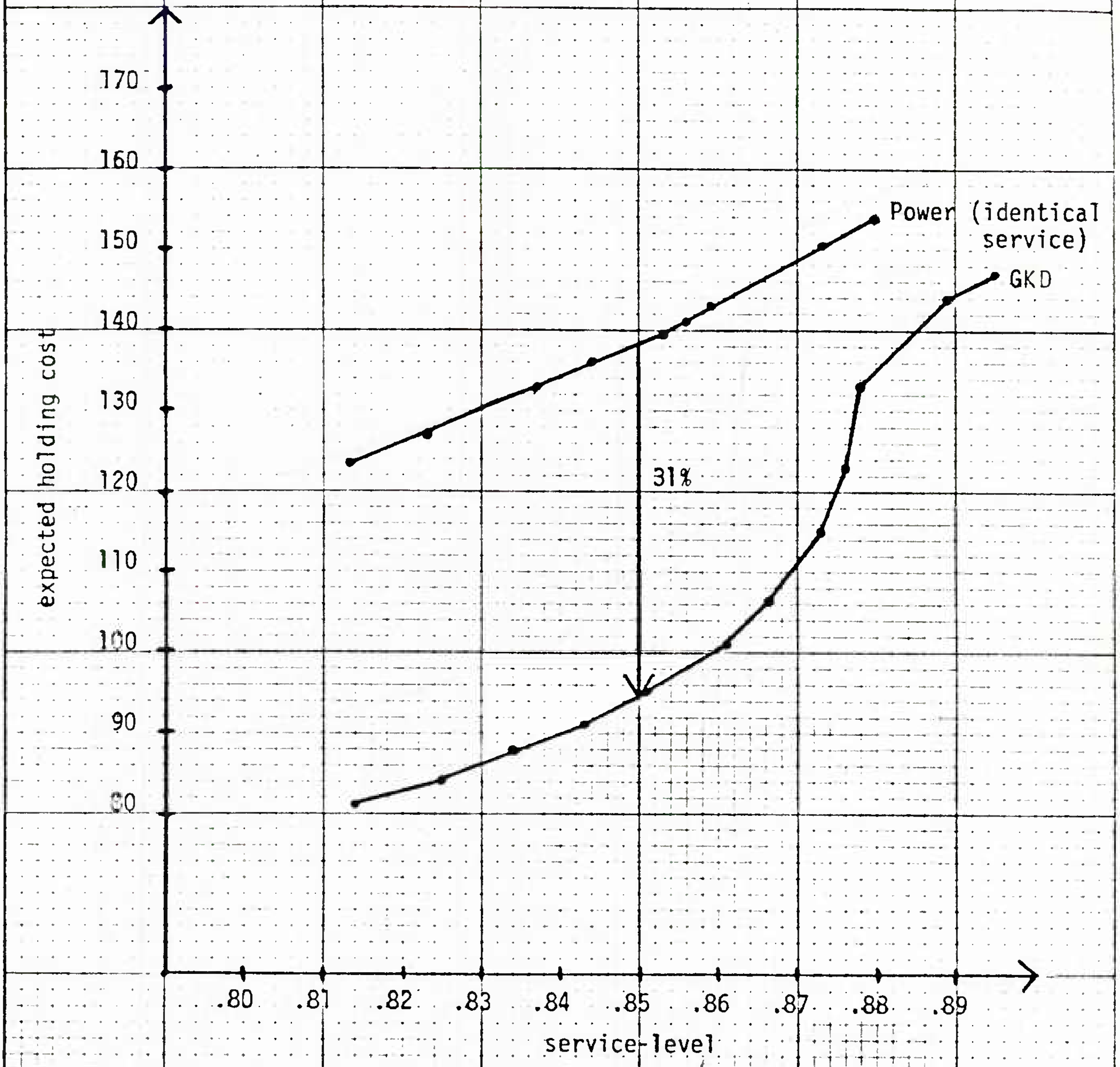


32 items, K_i=24, $W_i = \frac{1}{32}$, k_i=4,

$\frac{\sigma_i^2}{\mu_i}$ = 3, μ: 16, 8, 4, 2,...,16, 8, 4, 2

D_i using Power Method

Figure 6

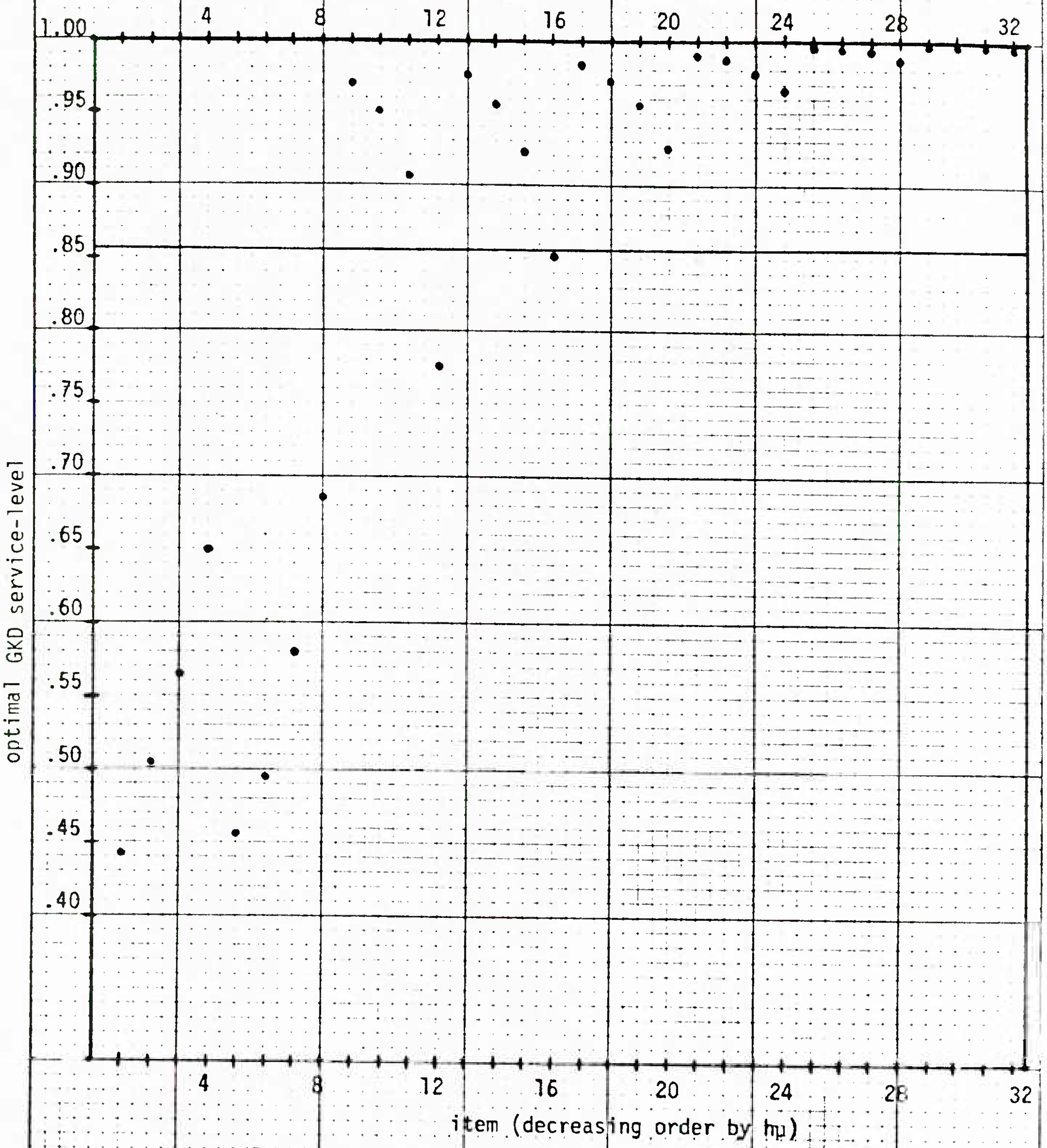


32 items, K_i=24, $W_i = \frac{1}{32}$, k_i=4,

$\frac{\sigma_i^2}{\mu_i}$ = 9, μ_i: 16,...,16, 8,...,8

D_i using Power Method

Figure 7

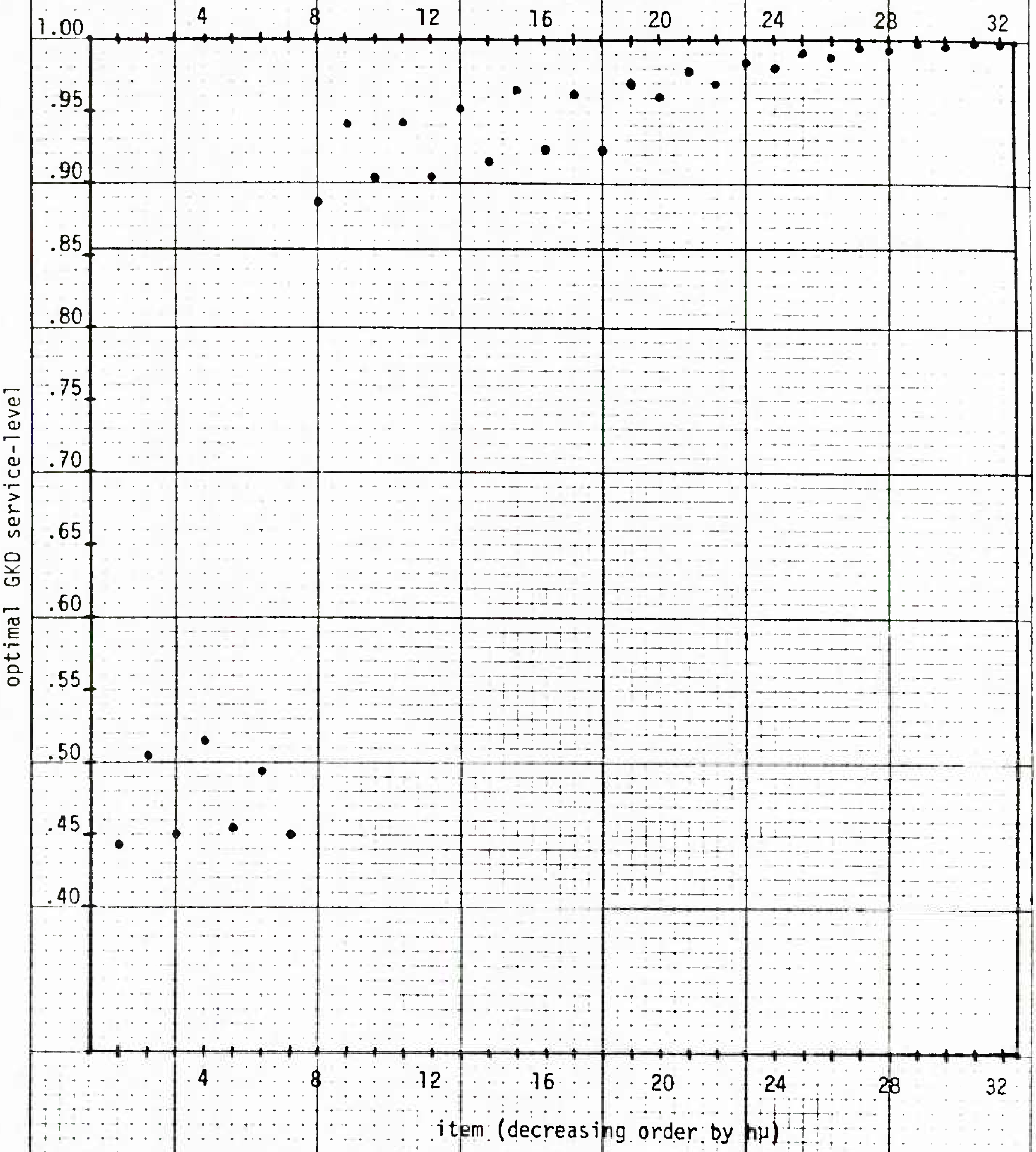


K_i=24, $W_i = \frac{1}{32}$, Average Service-Level=.856, k_i=4,

$\frac{\sigma_i^2}{\mu_i}$ = 9, μ_i: 16, 8, 4, 2,...,16, 8, 4, 2

D_i using Power Method

Figure 9

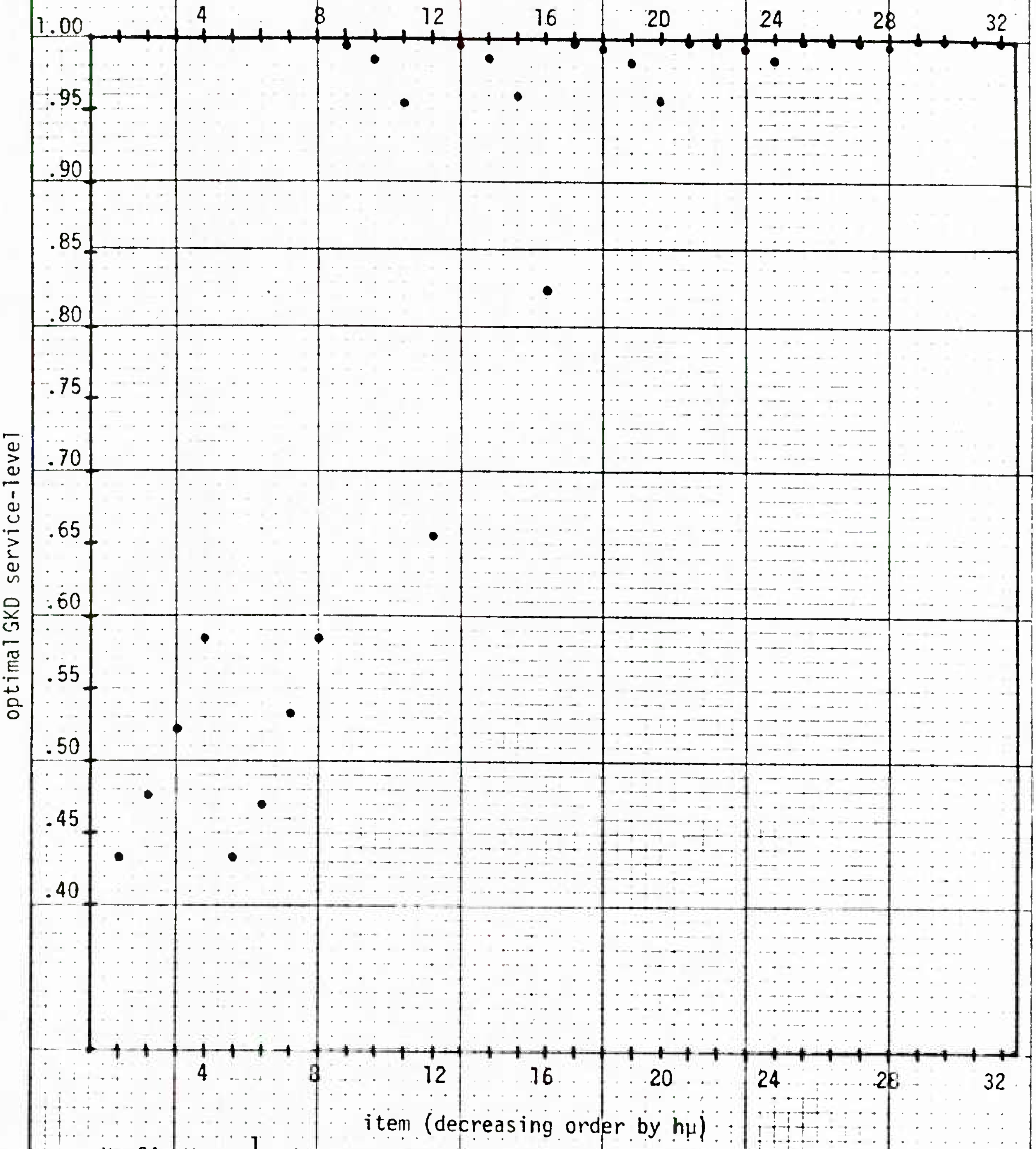


K_i=24, $W_i = \frac{1}{32}$, Average Service-Level = .854, k_i=4,

$\frac{\sigma_i^2}{\mu_i}$ = 9, μ_i: 16, 8,...,16, 8

D_i using Power Method

Figure 10

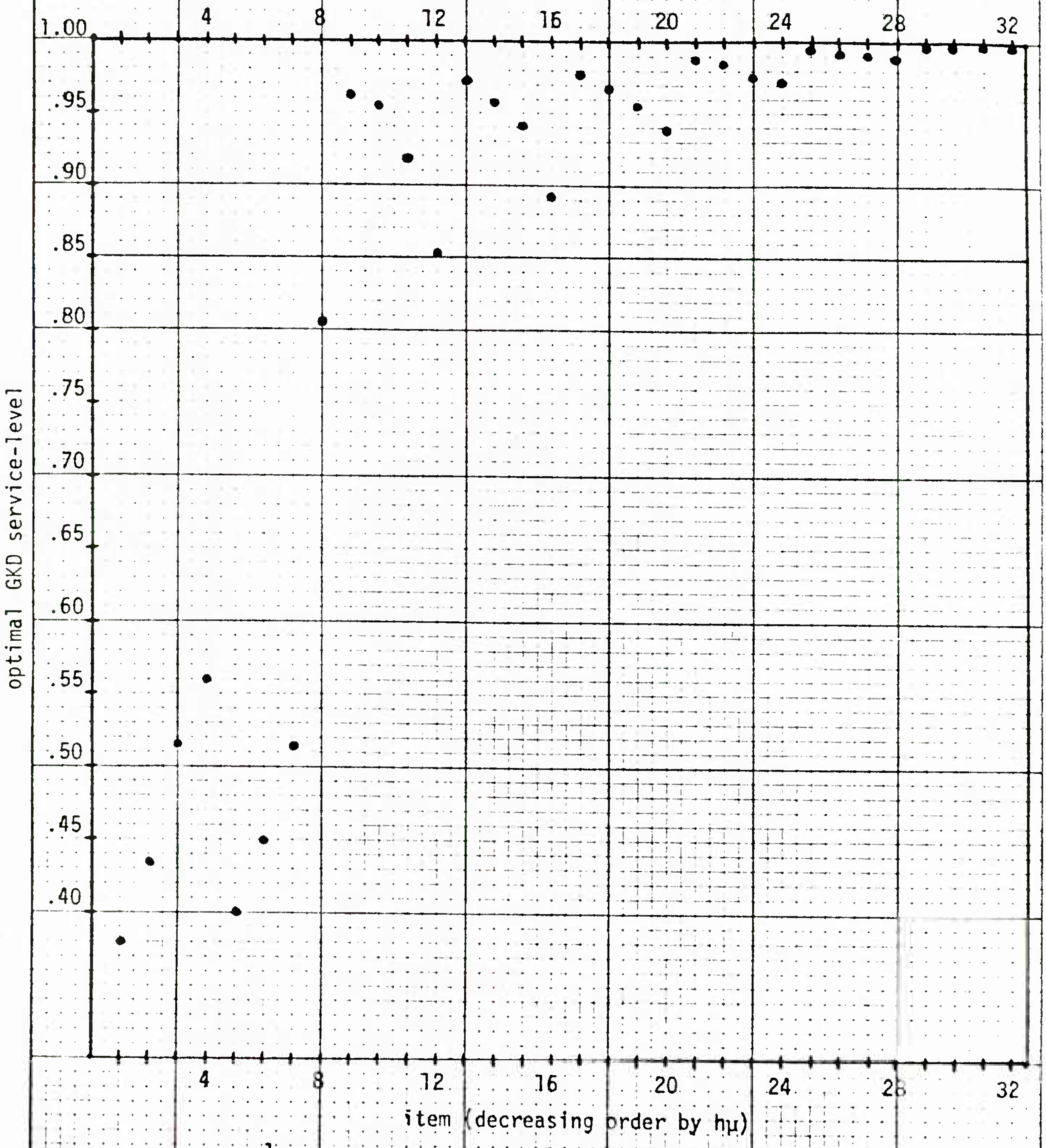


$K_i=24$, $W_i = \frac{1}{32}$, Average Service-Level = .852, $k_i=4$,

$\frac{\sigma_i^2}{\mu_i} = 9$, μ_i: 16, 8, 4, 2,...,16, 8, 4, 2

D_i using Power Method

Figure 11

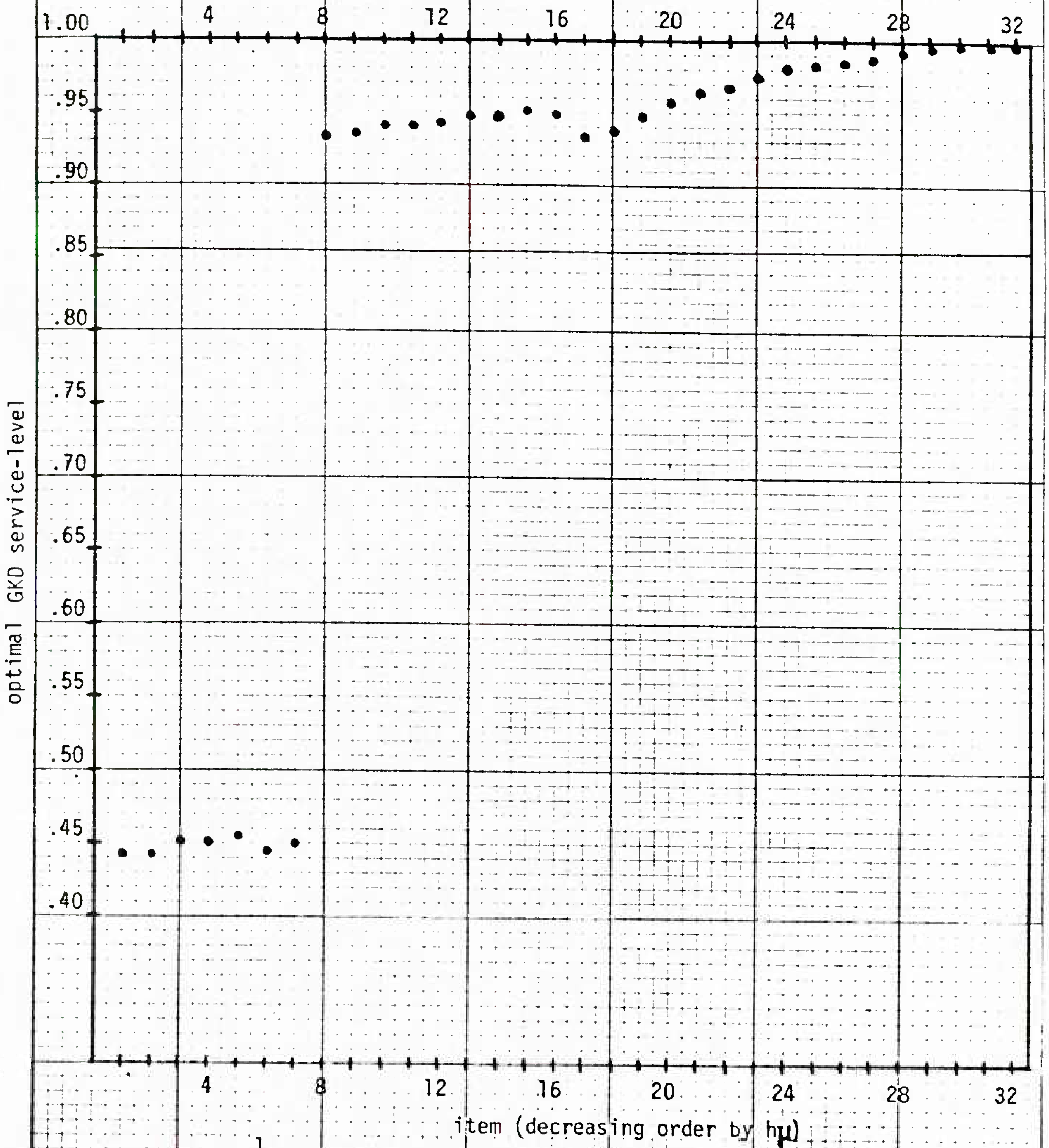


$K_i=24$, $W_i = \frac{1}{32}$, Average Service-level = .850, $k_1=4$,

$\frac{\sigma_i^2}{\mu_i} = 3$, μ_i: 16,...,16, 8,...,8

D_i using Power Method

Figure 12

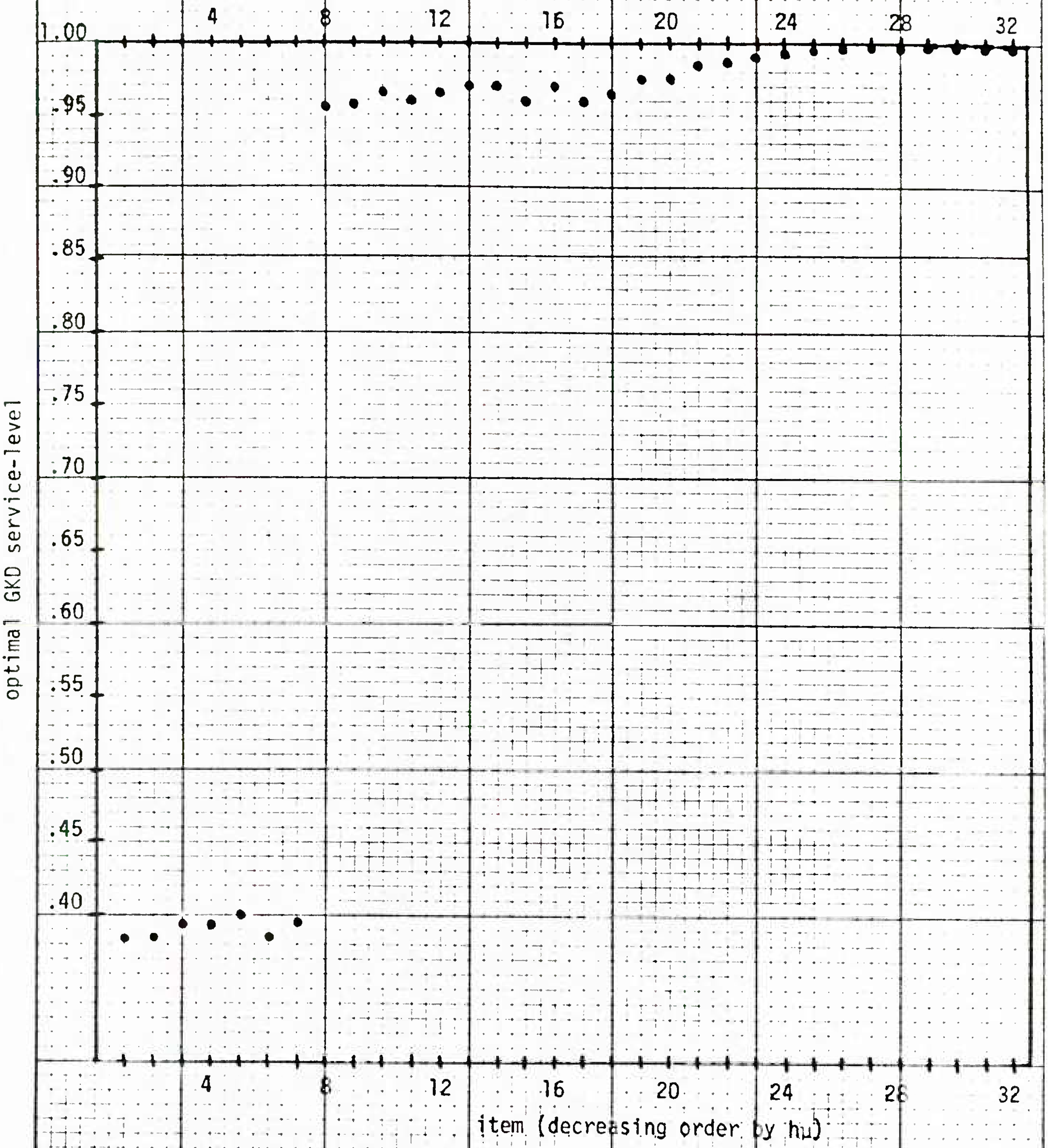


K_i=24, $W_i = \frac{1}{32}$, Average Service-level = .851, k_i=4,

$\frac{\sigma_i^2}{\mu_i} = 9$, μ_i: 16,...,16, 8,..., 8

D_i using Power Method

Figure 13



K_i=24, $W_i = \frac{1}{32}$, Average Service level = .851, k_i=8,

$\frac{\sigma_i^2}{\mu_i}$ = 3, μ_i: 16,...,16, 8,...,8

D_i using Power Method

Figure 14

Parameters, i=1,...,32			Figure with Cost Function	Figure with Service-levels	# items with $\xi_i>0$	items with $\xi_i>0$
k_i	$\frac{\sigma_i^2}{\mu_i}$	$\mu_1,\ldots,\mu_{32}$				
4	9	16,8,4,2,...	3	9	3	27,28,32
4	9	16,8,...	4	10	5	26,27,28,30,32
8	9	16,8,4,2,...	5	11	15	8,10-12,14,16,18-22,24,27,28,30
4	3	16,8,4,2,...	6	12	5	24,27,28,31,32
4	9	16,...,8,...	7	13	7	26-32
8	3	16,...,8,...	8	14	22	7-32

Table 2

For all systems: $K_i=24$, $i=1,\ldots,32$.

$W_i = \frac{1}{32}$, $i=1,\ldots,32$.

IV. SENSITIVITY EXPERIENCE WITH THE GKD ALGORITHM

1. INTRODUCTION

In Chapter II we considered the problem of specifying single-item service objectives in a multi-item (s,S) inventory system subject to a system-wide service objective. We formulated this problem as a constrained NLP and developed the Generalized Knapsack Duality (GKD) Algorithm to solve it. In Chapter III we reported our computational experience with several 32-item inventory systems with a structure typical of many real-world inventory systems. In this chapter we report sensitivity experience with the base-case and worst-case inventory systems used in Chapter III. The criterion of comparison is to compare the expected holding cost (computed exactly) associated with the (s,S) policies produced by the GKD Algorithm for a given system-wide service level α with the expected holding cost of the policies given by the Identical Service Method, in which all the (s,S) policies are set by the Power Approximation given in Mosier [1981] so that every item has a service-level α (as discussed in Chapter III the exact algorithm of Veinott and Wagner [1965] is computationally unwieldy for our inventory system, and the Power Approximation is the most accurate approximation available). In computing the expected costs we assume the underlying demand distribution is Negative Binomial, and we are primarily interested in the sensitivity of the GKD Algorithm at service-levels of 85% and 88%. We are interested in 85% because this is a service-level managers often prefer, and we are interested in 88% because of the difficulties reported in Chapter III with the worst-case inventory system at this service-level.

In Section 2 of this chapter we report sensitivity experience for $D_i=S_i-s_i$, which is a constant input parameter to the GKD Algorithm. In Section 3 we report sensitivity experience for the lower bounds on S_i, which are input to the GKD Algorithm, but may be increased by the GKD Algorithm (Chapter II Section 5). In Section 4 we consider several sampling schemes for large-scale inventory aggregation, where a sample is used with the GKD Algorithm to specify (s,S) policies for every item in the inventory system. In Section 5 we investigate the sensitivity of these sampling schemes to the number of items in the scheme, and in Section 6 we offer conclusions and recommendations concerning specifying the D_i, the lower bounds on S_i and the size and type of inventory scheme to use for large-scale inventory aggregation. All figures are at the end of the chapter.

2. SENSITIVITY EXPERIENCE FOR D_i

Recall from Section 2 of Chapter II that $D_i=S_i-s_i$ is approximated by a constant (the Power Approximation of D_i) which is input to the GKD Algorithm. This is justified because in the single-item case the optimal D_i is asymptotically independent of the service-level. For any given item i in the inventory system, however, the policy (s_i,S_i) produced by the GKD Algorithm with associated service-level α_i may not be an optimal policy when the item i is considered as a single-item inventory system. And for a non-optimal policy (s_i,S_i), D_i is not necessarily independent of the service-level α_i.

Thus we investigated the impact of varying the input values of the D_i, to determine if the GKD Algorithm performance degraded (or perhaps even improved). As stated in the Introduction, we used the 32-item base-case inventory system of Chapter III. This system has the structure that twenty percent of the items constitute eighty percent of the total value (where value is taken to be $h\mu$) of the inventory system. The parameters are as follows for i=1,...,N:

$$K_i = 24$$
$$W_i = \frac{1}{32}$$
$$k_i = 4$$
$$\frac{\sigma_i^2}{\mu_i} = 9$$

and

$$\mu_1,\mu_5,\mu_9,\ldots,\ \mu_{29} = 16$$
$$\mu_2,\mu_6,\mu_{10},\ldots,\ \mu_{30} = 8$$
$$\mu_3,\mu_7,\mu_{11},\ldots,\ \mu_{31} = 4$$
$$\mu_4,\mu_8,\mu_{12},\ldots,\ \mu_{32} = 2.$$

The h_i are determined through the valuc $h_i\mu_i$. Additional details are given in Chapter III.

Because of the structure of the inventory system, the items 1-25 are low-value items and items 26-32 are high-value items. Typically, as discussed in Section 3 of Chapter III, the GKD Algorithm produces policies so that the service-levels of the high-value items are very low and the service-levels of the low-value items are very high. These observations suggest two experiments.

In the first, we increased the value of D_i (given by the Power Approximation) 25% for the low-value items and decreased the value of D_i 25% for the high-value items. In the second experiment, we decreased the value of D_i by 25% for the low-value items and increased it 25% for the high-value items.

These are severe tests of robustness. In the experiments in Chapter III we compared the expected holding costs of the GKD Algorithm policies and the Power Approximation policies (which give identical service to all items) because, since we used the Power Approximation D_i in the GKD Algorithm, the expected replenishment costs are the same. Such is not the case in the experiments described in this section, so we compare the total expteced costs, namely, the expected holding plus replenishment costs. These experiments and their outcomes are summarized in Table 3 and Figures 15, 16 and 17. In Experiment 2, for example, in which the Power Approximation D_i is increased 25% for the low-value items and decreased 25% for the high-value items, the expected total cost for the GKD policies at a service-level of 85% is approximately 198, and the expected total costs for the Identical Service Method at that same service-level is approximately 276. This represents a 28% decrease in expected cost when using the GKD policies rather than the Identical Service policies.

Table 3. Sensitivity Experience for D_i: Comparison of GKD Algorithm Policies with Identical Service Policies.

Experiment	Low-Value Items (1-25)	High-Value Items (26-32)	% ↓ in cost at Service-Level 85%	% ↓ in cost at Service-Level 88%	See Figure
1	No change	No change	29	26	15
2	D_i ↑ 25%	D_i ↓ 25%	26	22	16
3	D_i ↓ 25%	D_i ↑ 25%	28	27	17

We make the following observations and conclusions:

(1) The change in the performance of the GKD Algorithm in Experiment 2 and 3 is slight, especially considering the severity of the test. This suggests that not only is an especially accurate specification of D unnecessary for a single-item inventory system, it is also unnecessary for a multi-item inventory system of the general structure of our base-case. Of course, additional experiments on variations of this base-case would help confirm or refute this conjecture.

(2) Although it may be possible to specify the D_i so that greater cost savings are realized than when using the Power Approximation D_i, we recommend using the Power Approximation, since obvious ways of changing D_i do not improve the GKD Algorithm performance.

3. SENSITIVITY EXPERIENCE FOR LOWER BOUNDS ON S_i

Recall from Chapter II that we required $S_i \geq D_i$ in the GKD Algorithm with that lower bound on S_i possibly being increased by the algorithm to guarantee convergence to a (possibly local) minimum (see Section 5). The main reason given to restrict attention to policies with $S_i \geq D_i$ is that (aside from the mathematical difficulties involved) many managers consider policies with $S_i < D_i$ (equivalently, $s_i < 0$) unsatisfactory for practical implementation. There are two reasons, however, that even higher lower bounds on S_i might be imposed.

First, many managers prefer not only that $s_i \geq 0$, but that $s_i \geq \mu_i^* = (k_i+1)\mu_i$, so the safety stock ($s_i$) is always at least large enough to cover the expected leadtime demand.

Second, in Chapter III the Identical Service Approach performed better than the GKD Algorithm in the worst-case experiment at a service-level at and near 88%.

We offer a conjecture to explain this behavior and show that by modifying the lower bounds on S_i the GKD Algorithm will in the case just described perform considerably better than the Identical Service Approach at both service levels 85% and 88%.

The worst-case inventory system has the same 80/20 structure (twenty percent of the items represent eighty percent of the total system value) with the following parameters for i=1,...,N:

$$K_i = 24$$

$$W_i = \frac{1}{32}$$

$$k_i = 8$$

$$\frac{\sigma_i^2}{\mu_i} = 9$$

and

$$\mu_1, \mu_2, \ldots, \mu_{16} = 16$$
$$\mu_{17}, \mu_{18}, \ldots, \mu_{32} = 8.$$

There is a notable difference in the single-item service levels determined by the GKD Algorithm at a service-level of only 85% for both the base-case and the worst-case inventory systems, as reported in Figures 9 and 14 of Chapter III, respectively. For the high-value items (items 1-7) in both cases the (s,S) policies given by the GKD Algorithm have all the S_i at their lower bounds. The service-levels associated with the high-value items for the worst-case are considerably lower than those for the base-case, and so the service-levels for the low-value items for the worst-case are considerably higher than those for the base-case. The GKD Algorithm assumes the demand distribution is Logistic, whereas in these experiments we assume the demand distribution is Negative Binomial. The approximation of a Negative Binomial distribution by a Logistic distribution is poorest in the tail, which corresponds to very high individual item service-levels. Thus we conjecture that the poor performance of the GKD Algorithm in the worst-case is due to the poorness of the Logistic demand approximation. We test this conjecture by imposing higher values of the lower bounds on the S_i. This forces the high-value items to be stocked at a higher service-level, so the low-value items are stocked at a lower service level, as in the base-case. We note that Figures 9 and 14 indicate that the higher the demand mean, the lower the service-level of the (s,S) policy with S at its lower bound.

perform as well at a service-level of 88% as it does at 91% with r=1. These experiments and their outcomes for the base-case are summarized in Table 6. The results are similar to those in Experiments 2, 3, and 4, and so we do not report the expected cost graphs.

Table 6. Sensitivity Experience for Lower Bounds on S_i: Base-Case: Comparison of GKD Algorithm Policies with Identical Service Policies.

Experiment	r, where $S_i \geq D_i + r\mu_i$*	Minimum Overall Service-Level	% ↓ in cost at Service-Level 88%
6	.8	.860	33
7	.9	.885	--

These experiments and their outcomes for the worst-case are summarized in Table 7 and Figure 22. Since the results of these experiments are similar to each other we report the expected cost graph only for Experiment 6.

Table 7. Sensitivity Experience for Lower Bounds on S_i: Worst-Case: Comparison of GKD Algorithm Policies with Identical Service Policies.

Experiment	r, where $S_i \geq D_i + r\mu_i$*	Minimum Overall Service-Level	% ↓ in cost at Service-Level 85%	% ↓ in cost at Service-Level 88%	See Figure
6	.8	.811	26	23	22
7	.9	.878	--	20	--

We make two additional observations:

(5) The previous remarks for the base-case also hold for r=.8, .9.

(6) For the worst-case system there is a dramatic improvement in algorithm performance when r=.8. In this case there is a substantial cost savings at both service-levels 85% and 88%.

We offer the following conclusions.

(1) The behavior at a service-level of 88% for the worst-case inventory system does seem to be caused (at least in part) by the poorness of approximating a Negative Binomial distribution by a Logistic distribution.

(2) The algorithm performs best when there is a lower bound on S just large enough to prevent too many of the low-value items to be stocked extremely high. We recommend that a user experiment with

various lower bounds (in terms of the single input value r) to find the best one.

(3) As most managers prefer as high a lower bound as possible on S (as long as average system service is not too large), we recommend specifying r as large as possible to include the desired targeted service-level.Our experience with both the best and worst-case inventory systems suggests that this approach will yield a significant cost reduction over the identical service approach.

4. SENSITIVITY EXPERIENCE FOR SAMPLING SCHEMES FOR INVENTORY AGGREGATION

Many real-world inventory systems contain tens of thousands of items. Although the GKD Algorithm is quite efficient, it could not handle inventory systems of this size. This is not a problem unique to our algorithm, and most managers deal with it using inventory aggregation. This is any method of partitioning an inventory system into groups of items so that every item in the group is controlled essentially the same. A representative item is chosen from each group and used to specify (s,S) policies for all items in its group.

Our method of aggregation exploits the special structure of our problem (as formulated as an NLP) and of the GKD Algorithm, and certain empirical observations in Chapter III. Recall from Section 3 of Chapter II that the GKD Algorithm exploits the "Knapsack" structure of the NLP by solving its dual NLP. The dual has only one variable, λ, and to find the optimal S^* of the original NLP it is sufficient to find the optimal dual variable λ^*. This is reflected in the GKD Algorithm (Chapter II

Section 3) by solving $\hat{P}(\lambda)=0$ for λ^*, from which S^* is computed. Our method of aggregation is to use a subset of the inventory system to estimate λ^*, and then compute S^* for all the items in the inventory system from this estimate.

The inventory system we constructed to test various aggregation schemes is a 128-item system with the same essential structure as the 32-item base-case system described in Section 2. In particular, twenty percent of the items (items 101-128) constitute eighty percent of the total value of the inventory system, and the parameters are as follows, for i=1, 2,..., 128:

$$K_i = 24$$

$$W_i = \frac{1}{128}$$

$$k_i = 4$$

$$\frac{\sigma_i^2}{\mu_i} = 9$$

and

$$\mu_1,\mu_5,\ldots,\mu_{125} = 16$$

$$\mu_2,\mu_6,\ldots,\mu_{126} = 8$$

$$\mu_3,\mu_7,\ldots,\mu_{127} = 4$$

$$\mu_4,\mu_8,\ldots,\mu_{128} = 2.$$

To examine the cost savings in using the GKD Algorithm rather than the Identical Service Method (via the Power Approximation) described in Section 1, we ran the GKD Algorithm with the entire system. As described in Section 1, we compared the expected holding costs with those of the Identical Service Method, and the result is displayed in Figure 23.

divided by the total value of the sample from that strata. This was done for both strata, so each maintained its total value, thus preserving the 80/20 structure. And to maintain the given constraint weights W_i, they were each recomputed as the total weight of the strata divided by the total weight of the sample from that strata.

Initially we tested three central nonrandom stratified uniform sampling schemes. If one wants to choose M items from a stratum consisting of the N items 1, 2,..., N by such a scheme, then the items

$$\frac{N}{2M}, \frac{3N}{2M}, \frac{5N}{2M}, \ldots, \frac{(2M-1)N}{2M}$$

are taken, where the fractions $\frac{jN}{2M}$ are rounded to the nearest integer. For this sample, the 80/20 value structure and the constraint-weight structure is maintained by multiplying h_i and K_i by $\sum_{\substack{i \text{ in} \\ \text{system}}} h_i \mu_i / \sum_{\substack{i \text{ in} \\ \text{sample}}} h_i \mu_i$, and W_i by N/M for each item in the sample. The three schemes differed in the number of items taken from each of the two strata. They are described in Table 8. Notice that Scheme 1 is a nonstratified scheme that roughly takes every 4th item from the inventory system (since the means repeat themselves every four items, to take every 4th item would create an inventory system with every mean the same, which seemed objectionable). Scheme 2 creates an inventory subsystem of half high-value and half low-value items, and that Scheme 3 takes all the high-value items (it seems reasonable that λ^* is more sensitive to them than to the very low value items, and so this scheme seemed plausible).

Table 8. Central Nonrandom Stratified Uniform Sampling.

Scheme	Number Chosen from Low-Value (1-100)	Number Chosen from High-Value (101-128)
1	24	8
2	16	16
3	4	28

The experiment we performed to test these schemes is as follows: For a variety of specified service-levels we ran the GKD Algorithm using the inventory subsystem (determined by the sampling scheme) to compute $\hat{\lambda}^*$, the optimal multiplier for the subsystem. Each value of $\hat{\lambda}^*$ gives rise to a service-level for the subsystem (α_{32}) and for the entire system (α_{128}), and to the expected holding cost at these service-levels (H_{32} and H_{128}, respectively). The criterion of comparison is how close α_{32} and α_{128} and H_{32} and H_{128} (for a given $\hat{\lambda}$) are. A "perfect" subsystem would give rise to $\alpha_{32}=\alpha_{128}$, $H_{32}=H_{128}$, that is, its optimal multiplier $\hat{\lambda}^*=\lambda^*$, the optimal multiplier for the entire system. The service-level graphs for the three schemes are given in Figures 24, 25 and 26, and the expected holding cost graphs are given in Figures 27, 28 and 29.

We make the following observations concerning these sampling schemes:

(1) They are all good. Although each one gives rise to a misspecification of α_{128} and H_{128}, the total cost savings will still be

substantial over the Identical Service Approach (compare Figure 23).

(2) Scheme 2 seems to be superior to Schemes 1 and 3. Scheme 1 has a preponderance of low-value items, Scheme 3 has a preponderance of high-value items, and Scheme 2 has the same number of each. It seems that the proportion of items that are high-value/low-value at least in part affects the performance of the scheme.

The efficiency of Scheme 2 may have something to do with the fact that the demand means of our base-case inventory system follow the pattern 2, 4, 8, 16,..., 2, 4, 8, 16. To investigate this, we used a central random stratified uniform sampling scheme with 16 expensive and 16 inexpensive items (like Scheme 2). In other words, we chose 16 items uniformly randomly (using the IMSL Routine GGUD) from both the high-value and low-value items. We performed six such tests, one with a different initial seed. The details of the subsystems chosen can be found in the Appendix to this chapter, and the scheme itself, Scheme 4, is described in Table 9.

Table 9. Central Random Stratified Uniform Sampling

Scheme	Number Chosen from Low-Value (1-100)	Number Chosen from High-Value (101-128)
4	16	16

We performed the same experiment as we did for the first three schemes, and the results of this experiment for the first seed is reported in Figures 30 and 31 (results for all of the seeds were similar). We make the following observations.

(1) As for the three nonrandom schemes, the random scheme is good.

(2) The random scheme is not as good as Scheme 2, the nonrandom scheme with the same proportion of high-value/low-value items as this scheme. It would appear that the sequence of demand means is important, and that further research in stratifying not only on the value but on the demand means could give even better sampling schemes.

(3) The method of sampling seems relatively unimportant as long as the 80/20 value structure of the inventory system is maintained.

We recommend, therefore, that the scheme be chosen on the basis of its ease in implementation. Perhaps the nonrandom Scheme 1 is easiest, as it requires only one stratum (it represents a central sampling of every 4th item).

As a final experiment we tested whether Scheme 4 with the seed of the first experiment was accurate when lower bounds are imposed on the S_i. We chose Scheme 4 because it gave rise to neither the best nor the worst performance among the schemes. We chose lower bounds corresponding to r=.75 (see Section 3) because this was the largest value of r which allowed an overall service-level of at least 85%. This experiment is described in Table 10 and the outcome is reported in Figures 32, 33, and 34, where Figure 32 reports the expected cost when using all 128 items in the GKD Algorithm.

Table 10. Central Random Stratified Uniform Sampling: $S_i \geq D_i + .75\mu_i^*$.

Scheme	Number Chosen from Low-Value (1-100)	Number Chosen from High-Value (101-128)
4	16	16

We make the following observations:

(1) There is a significant cost savings at service-levels 85% and 88%.

(2) The sampling scheme is very accurate. It seems that our previous conclusions about the accuracy of these sampling schemes also hold true when lower bounds are imposed on the S_i.

5. SENSITIVITY EXPERIENCE FOR THE SAMPLE SIZE FOR INVENTORY AGGREGATION

In order to investigate how the reliability of inventory sampling schemes depends on the size of the scheme, we constructed a 512-item base-case inventory system with the 80/20 value structure described in Section 2. Not only is this structure typical of many real-world inventory systems, but if the items in the system are ordered in descending order by their value ($h_i\mu_i$) and the cumulative percent of the items is plotted against the corresponding cumulative value, then the resulting graph is frequently well-approximated by a Lognormal cumulative distribution function (Peterson and Silver [1979, pp. 30-37]). We constructed

(Nonrandom) Uniform Sampling Scheme (compare Scheme 1 of the previous section). The last experiment (Scheme 4) of Section 4 suggests that the approach of using a sample to investigate the sensitivity of the system to r is a reliable procedure. We considered r=0, .6, .7, .8, .9, 1, with a targeted service-level of 85% (a service-level popular with many managers), and we want to include for consideration service-levels 83% to 88%.

This experiment and its result is reported in Table 11 and Figure 35. Since we want to include for consideration a service-level of 83% we chose r=.6. The cost savings for r=.6 at a service-level of 85% is about 51%, which is very signficant.

Table 11. 128-Item Central Nonrandom Uniform Sample from the 512-Item Base-Case System: Comparison of GKD Algorithm Policies with Identical Service Policies.

r, where $S_i \geq D_i + r\mu_i$*	Minimum Overall Service-Level	% ↓ in cost at Service-Level 85%
0	< .8100	50
.6	.8140	51
.7	.8439	47
.8	.8719	--
.9	.8951	--
1	.9163	--

As an additional check of the reliability of using this sampling method to chose r, we repeated the experiment on the entire 512-item inventory system. The results are summarized in Table 12 and Figure 36. The sampling method of choosing r is seen to be very reliable, as was the case in Section 4.

Table 12. 512-Item Base-Case System: Comparison of GKD Algorithm Policies with Identical Service Policies.

r, where $S_i \geq D_i + r\mu_i$*	Minimum Overall Service-Level	% ↓ in cost at Service-Level 85%
0	< .8100	49
.6	.8138	50
.7	.8441	46
.8	.8720	--
.9	.8961	--
1	.9165	--

There were two sampling schemes considered in Section 4 which seemed to perform the best. The first is the recommended scheme, the Central Nonrandom Uniform Sampling Scheme. The second is the Central Nonrandom

Stratified Uniform Sampling Scheme (compare Scheme 2) in which half the sample is taken from the high-value items and the other half is taken from the low-value items. Because of the 80/20 value structure of our 512-item base-case system, items 1-102 are high-value and items 103-512 are low-value. The sensitivity experimental design is identical to that described in Section 4, here with r=.6. We chose sample sizes of 32, 64 and 128 for both schemes described, and the six experiments are summarized in Table 13. For each sample size, the results of these experiments were similar for both the uniform and stratified uniform schemes, and so the comparative graphs are reported only for the Central Nonrandom Uniform Sampling Scheme (Figures 37 through 42).

Table 13. Comparison of GKD Algorithm Policies with Identical Service Policies.

Central Nonrandom Uniform Sampling from 512-Item Base-Case System, $S_i \geq D_i + .6\mu_i^*$.

Experiment	Sample Size	Figure Reporting Service-Levels	Figure Reporting Expected Costs
1	32	37	40
2	64	38	41
3	128	39	42

Central Nonrandom Stratified Uniform Sampling from 512-Item Base-Case System, $S_i \geq D_i + .6\mu_i^*$.

Experiment	Number from High-Value Items	Number from Low-Value Items
4	16	16
5	32	32
6	64	64

We make the following observations:

(1) All six schemes are reliable enough for practical implementation. The 128-item schemes are superior to the 64-item schemes which are superior to the 32-item schemes.

(2) There does not seem to be any real difference between the uniform and stratified schemes.

(3) An examination of the experiments in Section 5. in which 32-item samples were taken from a 128-item inventory system, indicates that the 32-item samples are just as reliable in the 512-item system as in the 128-item system. This is extremely significant because it strongly suggests that 32-item schemes are sufficiently reliable for practical implementation no matter how large the inventory system is, as long as it has the 80/20 value structure described.

(4) We recommend using the Central Uniform Sampling Scheme with around 32 items for any inventory system with an 80/20 value structure. Although larger samples may be more reliable, this scheme is reliable enough. Moreover, the computational work necessary to use the GKD Algorithm on such a sample is very small, making this scheme suitable for frequent use in a real-world setting.

6. CONCLUSIONS

In this chapter we performed sensitivity tests on the GKD Algorithm with inventory systems with a structure typical of real-world systems, that twenty percent of the items represent 80 percent of the value of the system. We concluded the following:

(1) The total cost savings are not affected very much by changes in D_i.

(2) Increasing the lower bounds on S_i may improve the algorithm performance, especially when the high-value items also tend to have high expected demands. Even when increasing these lower bounds degrades the performance, however, the cost savings are substantial. We recommend increasing these lower bounds as high as possible (while still maintaining a cost savings below that of the Identical Service Method) as this appraoch is most consistent with managerial goals.

(3) Uniform sampling with about 32 items is a reliable and computationally efficient method of inventory aggregation, giving accurate predictions via the GKD Algorithm of overall inventory service and expected operating costs.

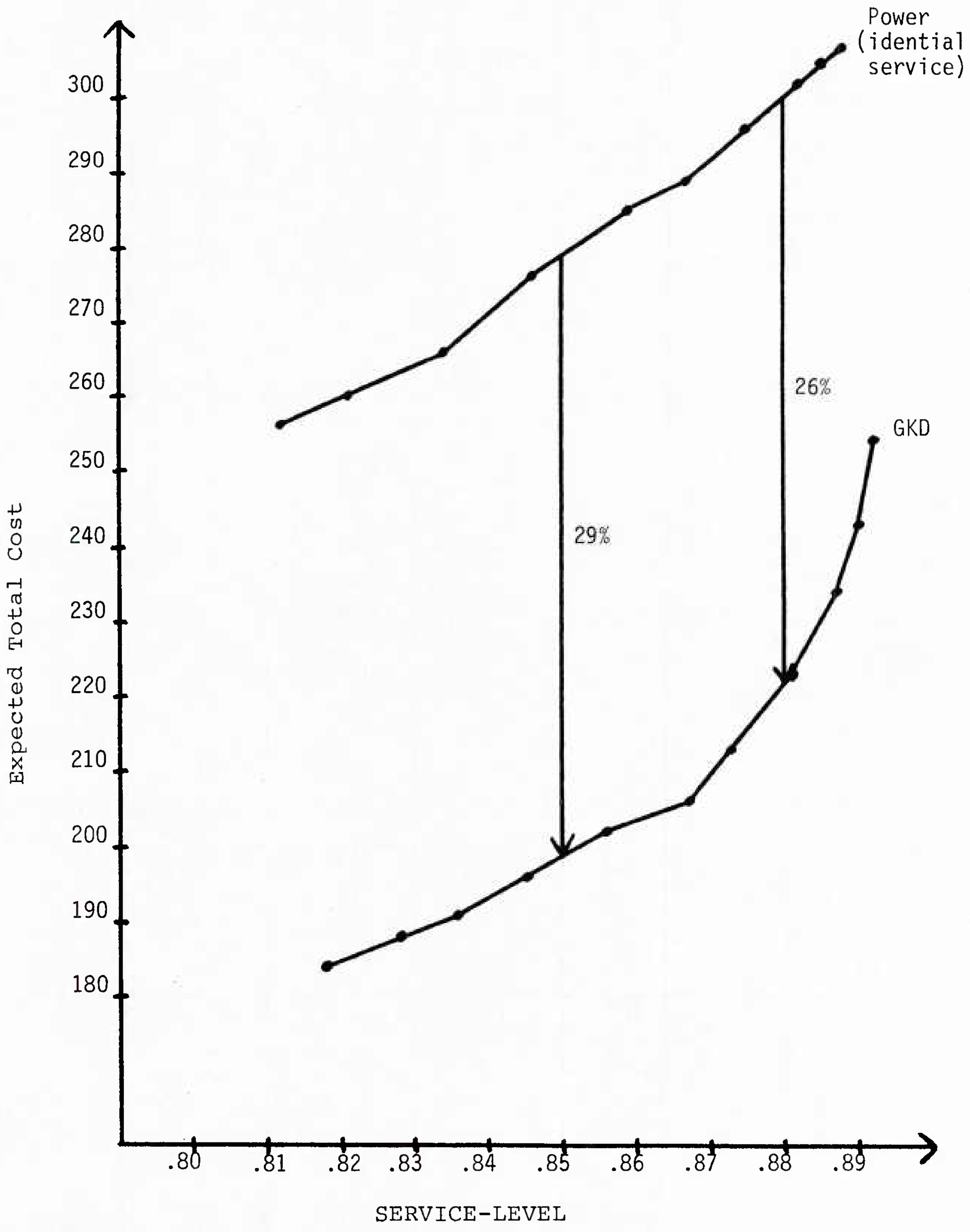


Figure 15. 32-Item Base - Case

D_i using Power Method

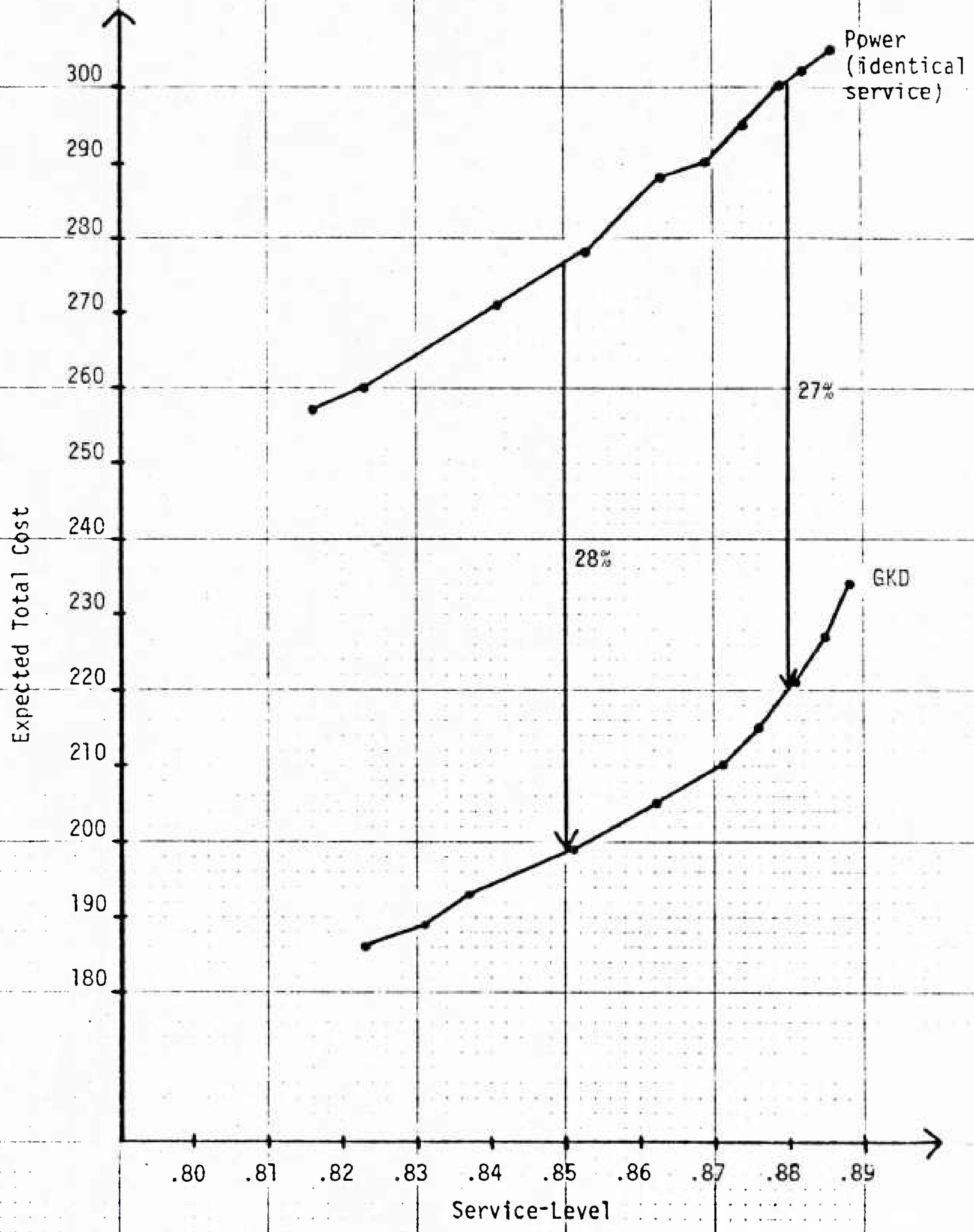


Figure 16. 32-Item Base-Case

D_i ↑ 25% for Low-Value Items

D_i ↓ 25% for High-Value Items

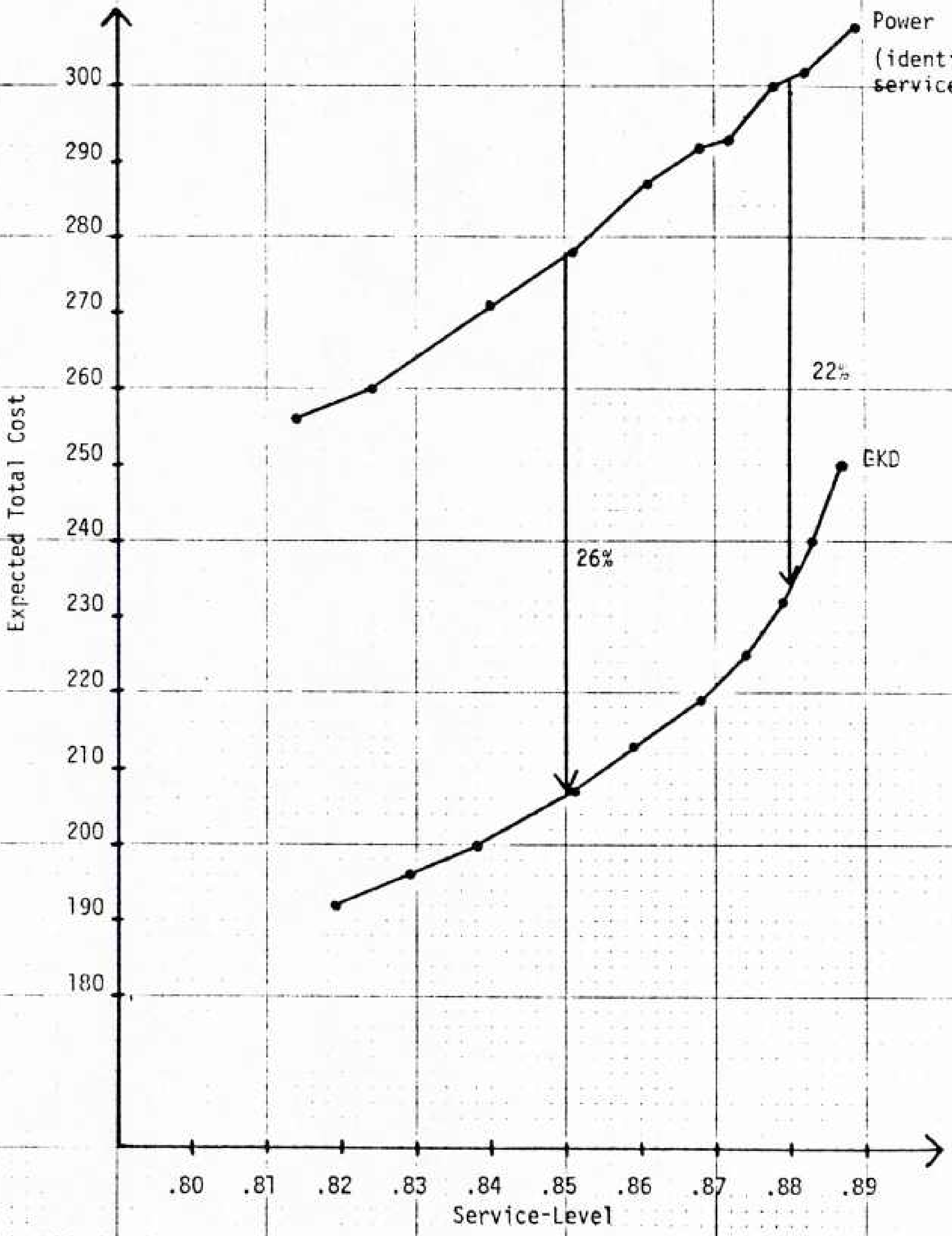


Figure 17. 32-Item Base-Case

$D_i \uparrow$ 25% for Low-Value Items

$D_i \downarrow$ 25% for High-Value Items

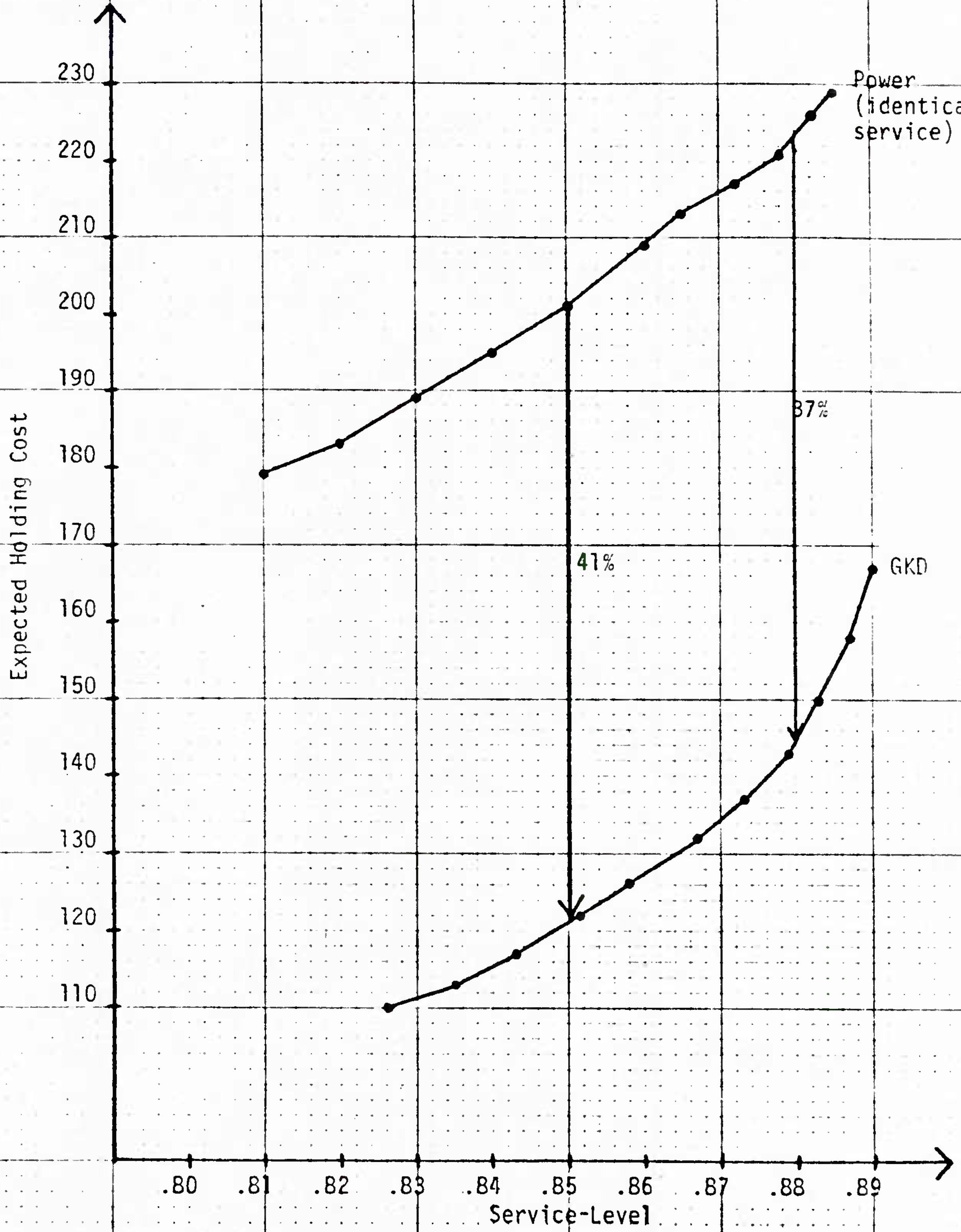


Figure 18 32-Item Base-Case

$S_i \geq D_i + .25\mu_i$*

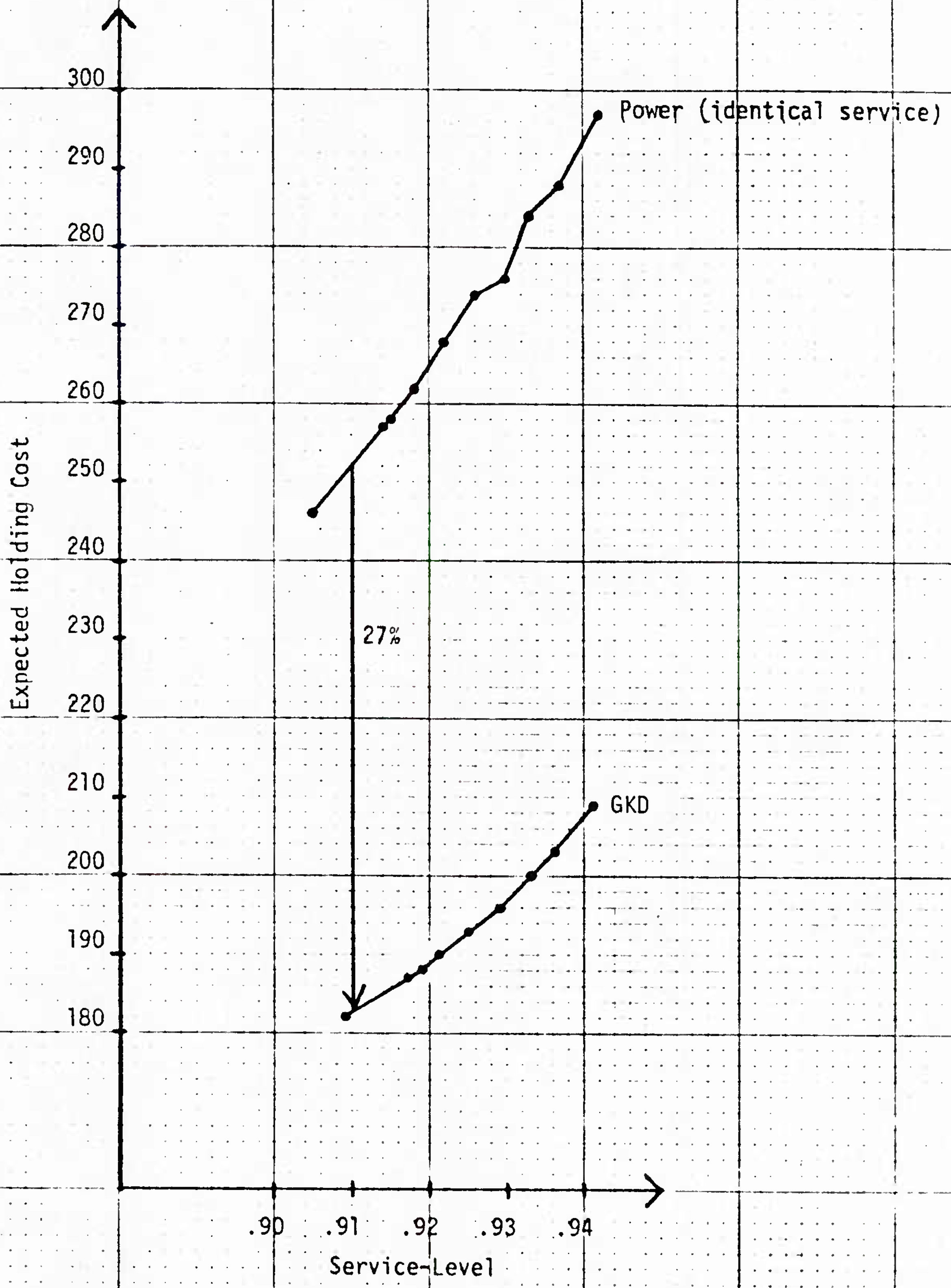


Figure 19 32-Item Base-Case

$S_i \geq D_i + \mu_i^*$

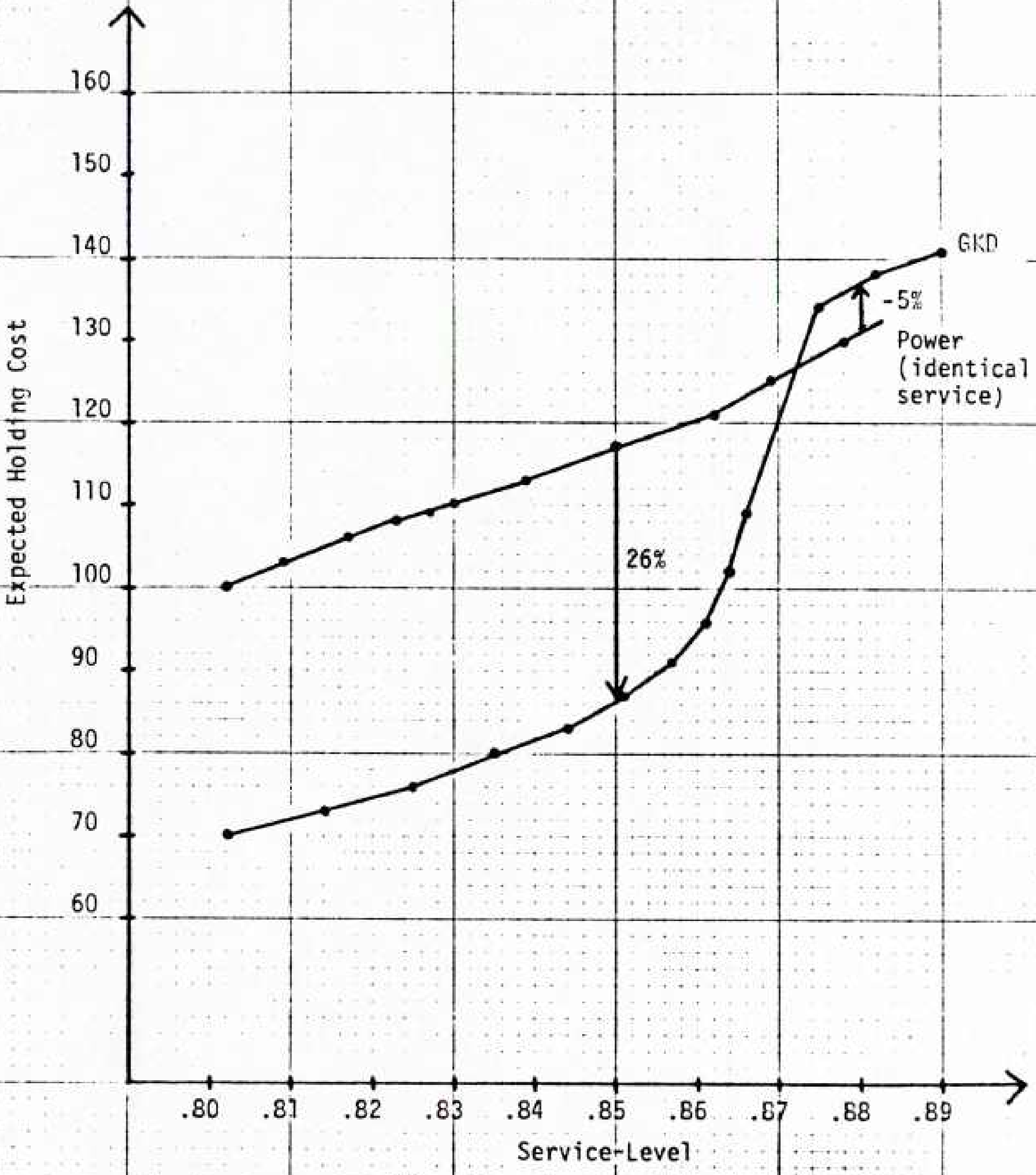


Figure 20 32-Item Worst-Case

$S_i \geq D_i + .25\mu_i$

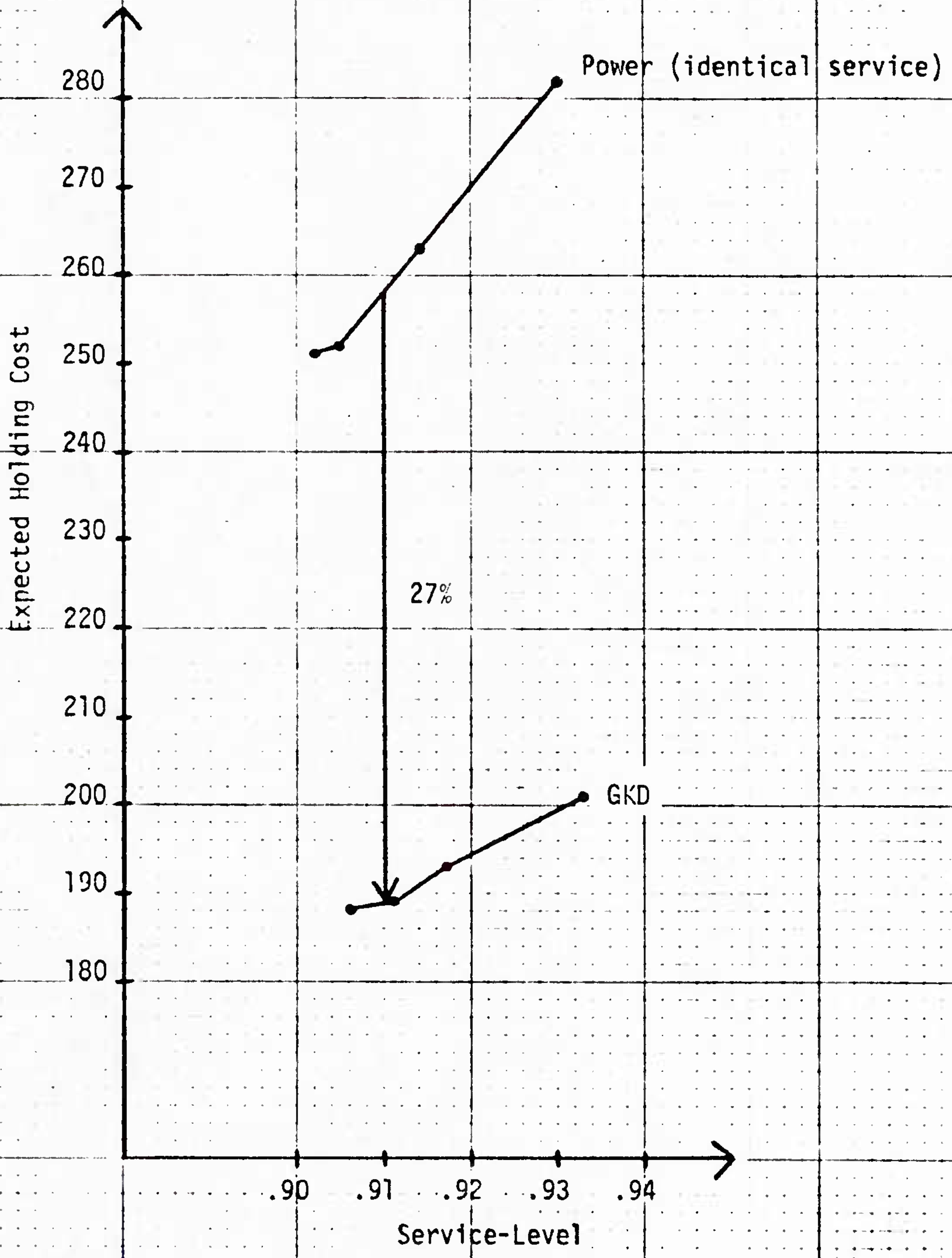


Figure 21. 32-Item Worst-Case

$S_i \geq D_i + \mu_i^*$

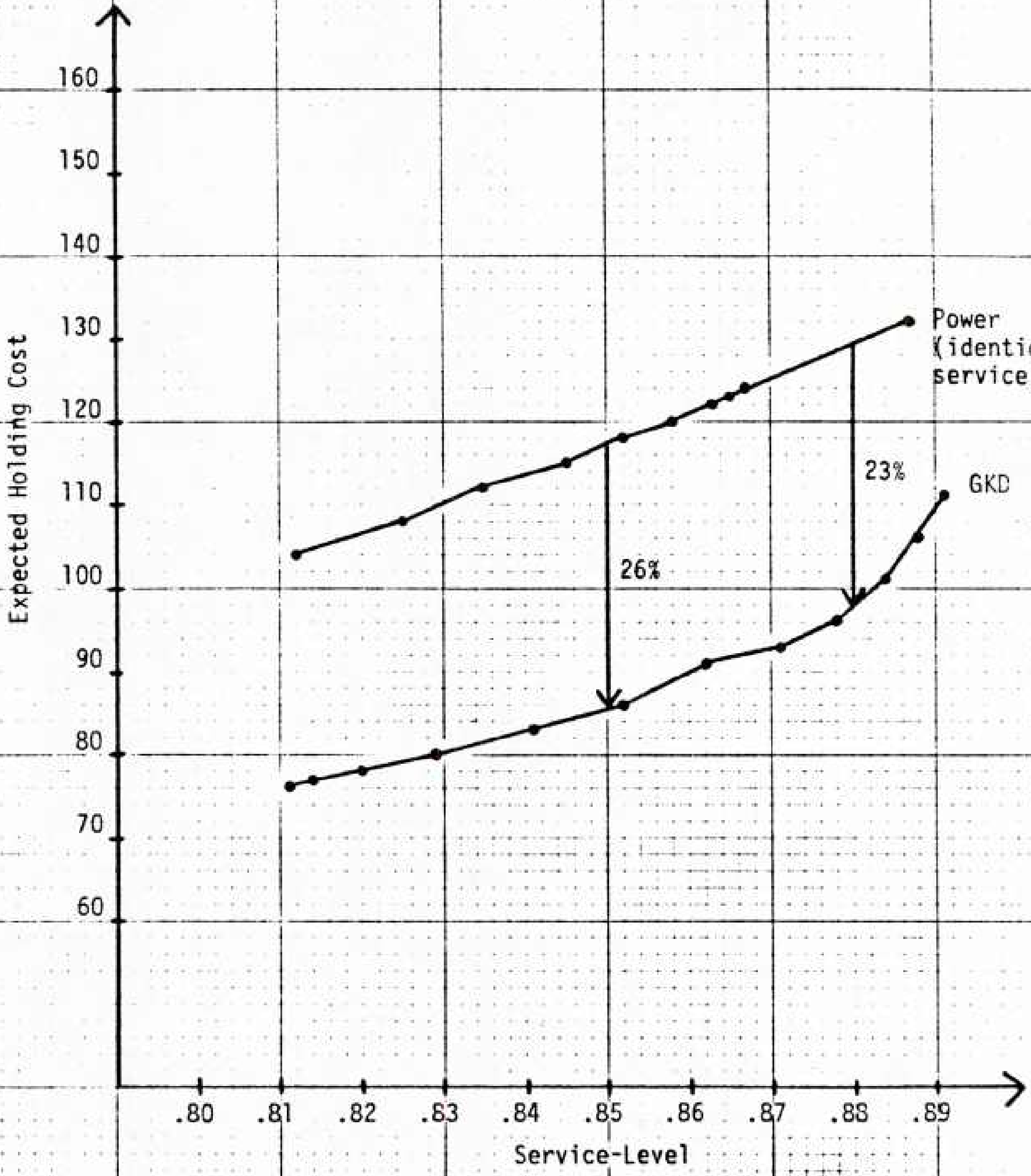


Figure 22. 32-Item Worst Case

$S_i \geq D_i + .80\mu_i^*$

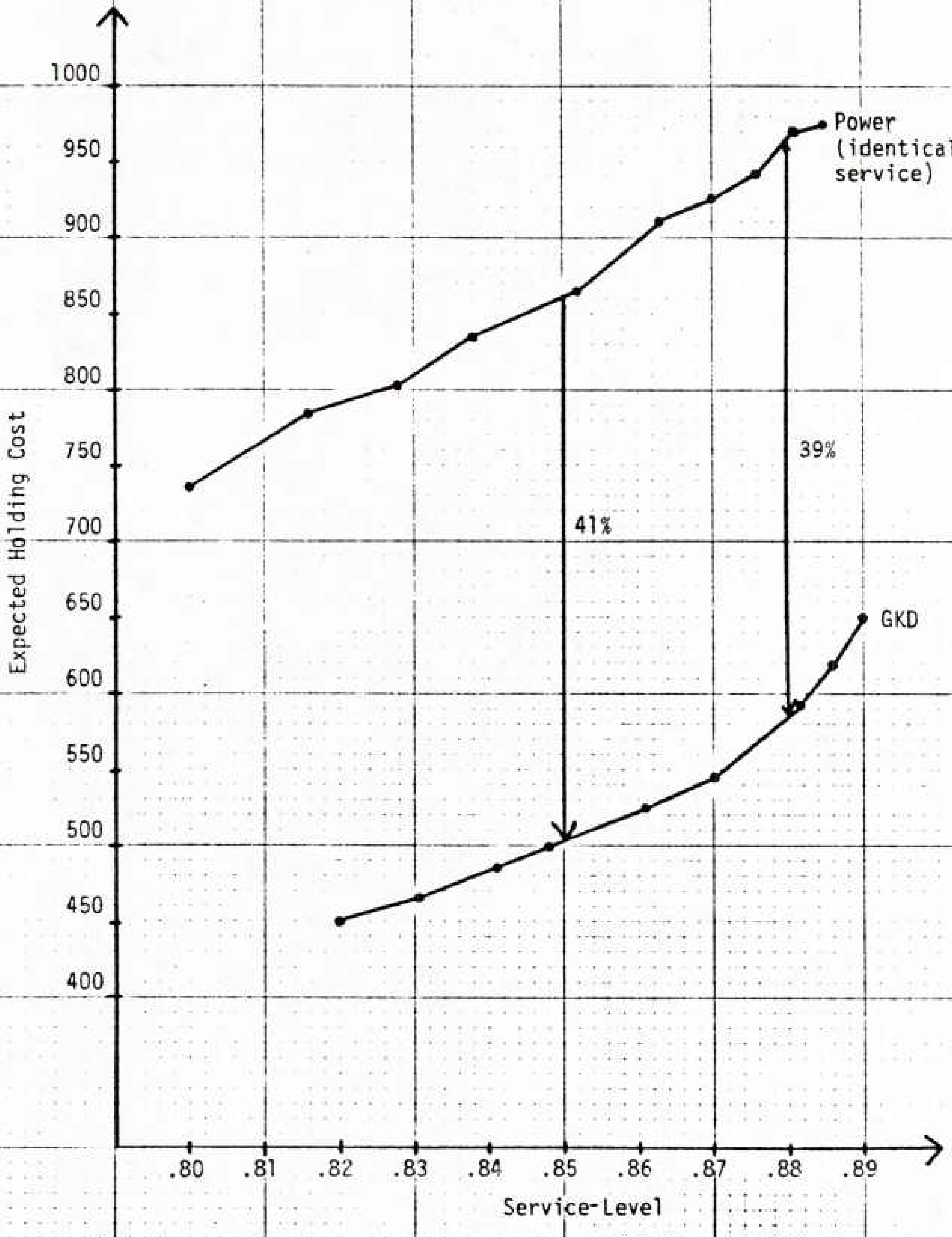


Figure 23. 128-Item Base-Case

$S_i \geq D_i$

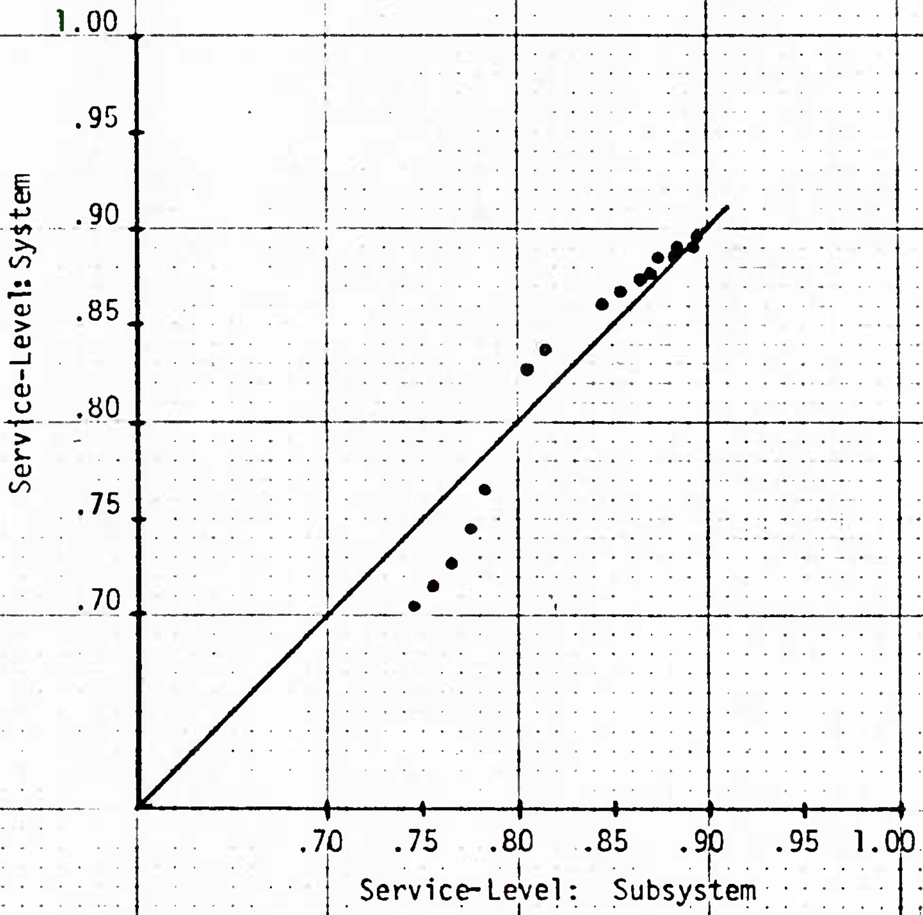


Figure 24. 128-Item Base-Case with Central Nonrandom Stratified Uniform Sampling Scheme 1 $S_i \geq D_i$

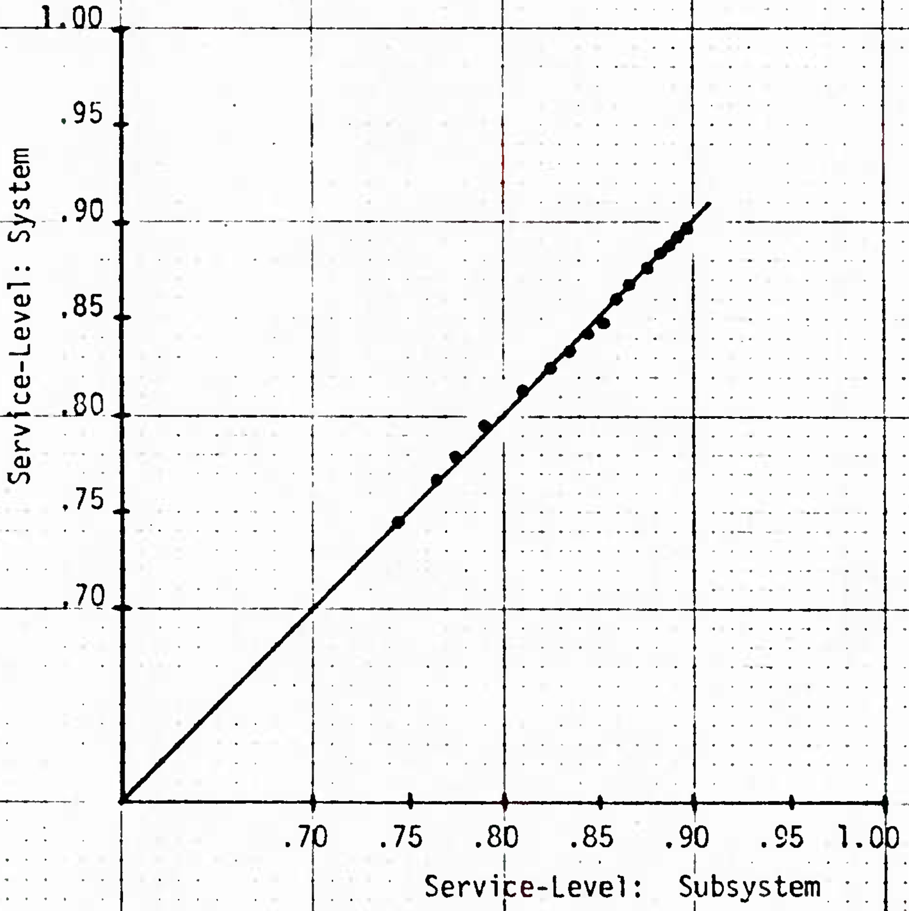


Figure 25. 128-Item Base-Case with Central Nonrandom Stratified Uniform Sampling Scheme 2 $S_1 \geq D_1$

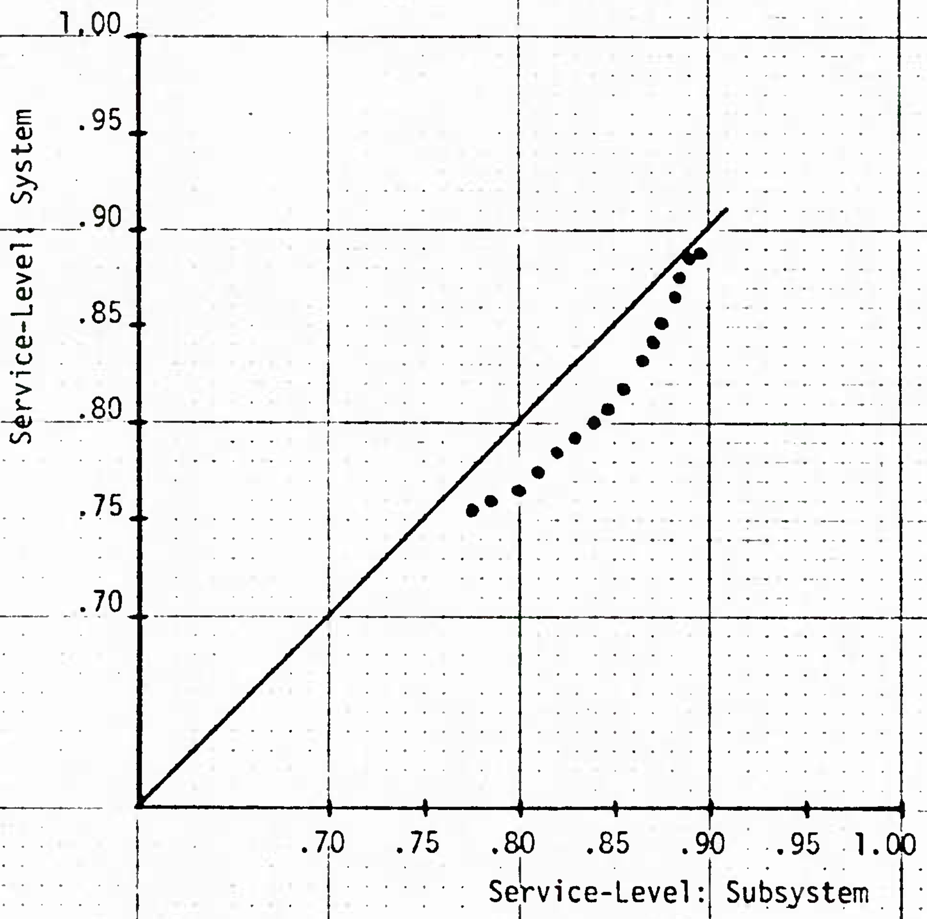


Figure 26. 128-Item Base-Case with Central Nonrandom Stratified Uniform Sampling Scheme 3
$S_i \geq D_i$

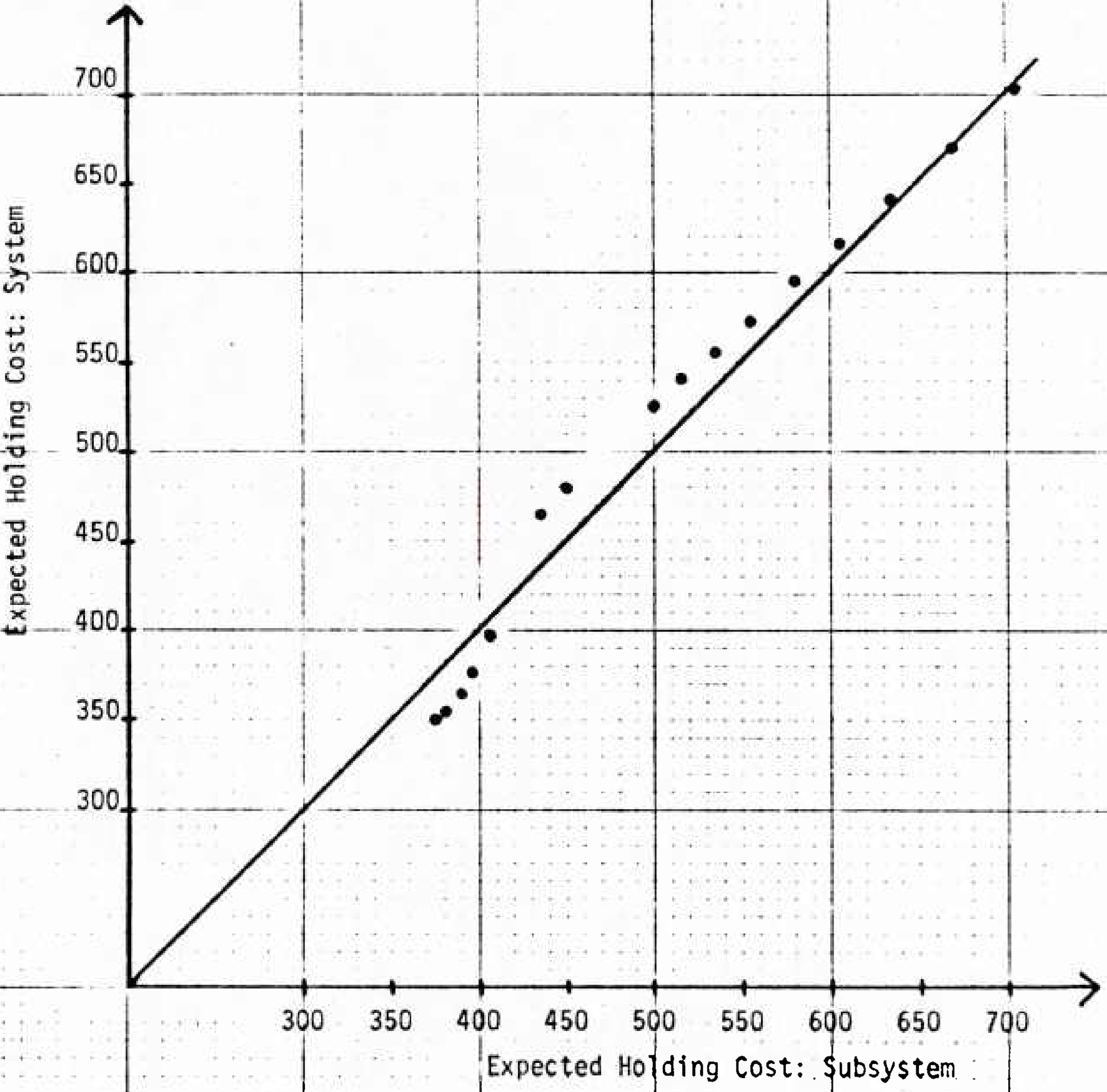


Figure 27. 128-Item Base-Case with Central Nonrandom Stratified Uniform Sampling Scheme 1

$S_i \geq D_i$

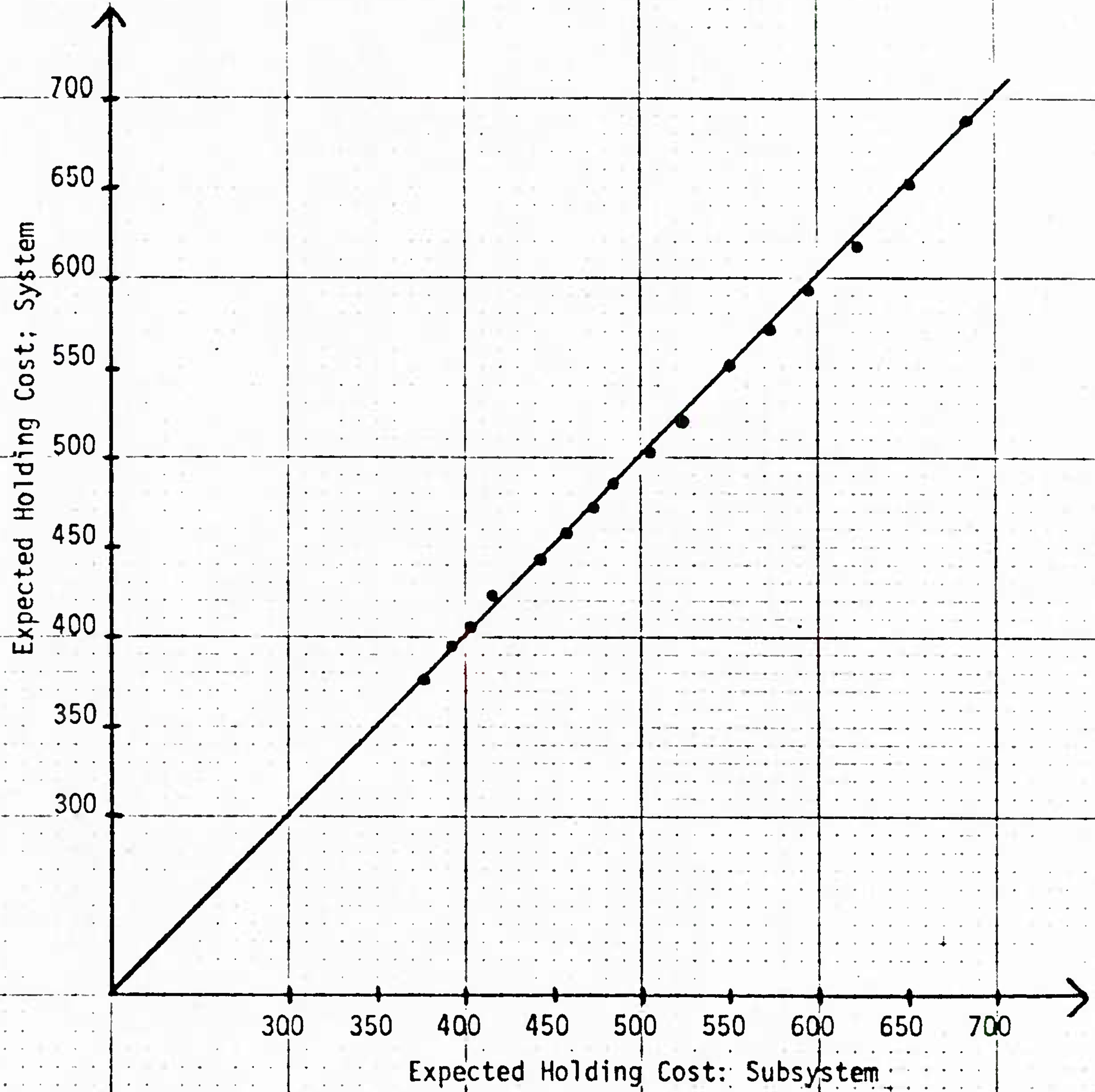


Figure 28. 128-Item Base-Case with Central Nonrandom Stratified Uniform Sampling Scheme 2

$S_i \geq D_i$

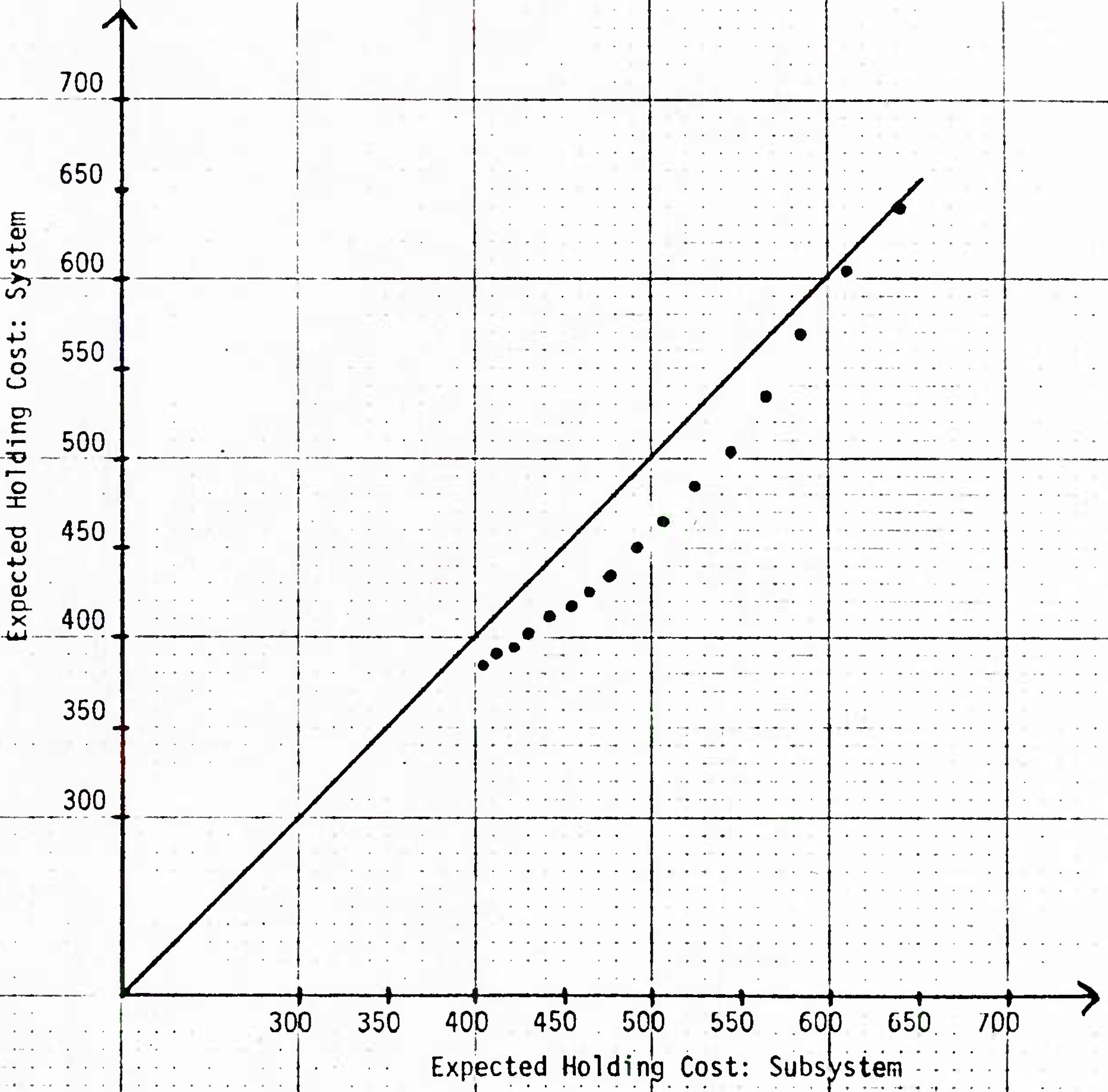


Figure 29. 128-Item Base-Case with Central Nonrandom Stratified Uniform Sampling Scheme 3

$S_i \geq D_i$

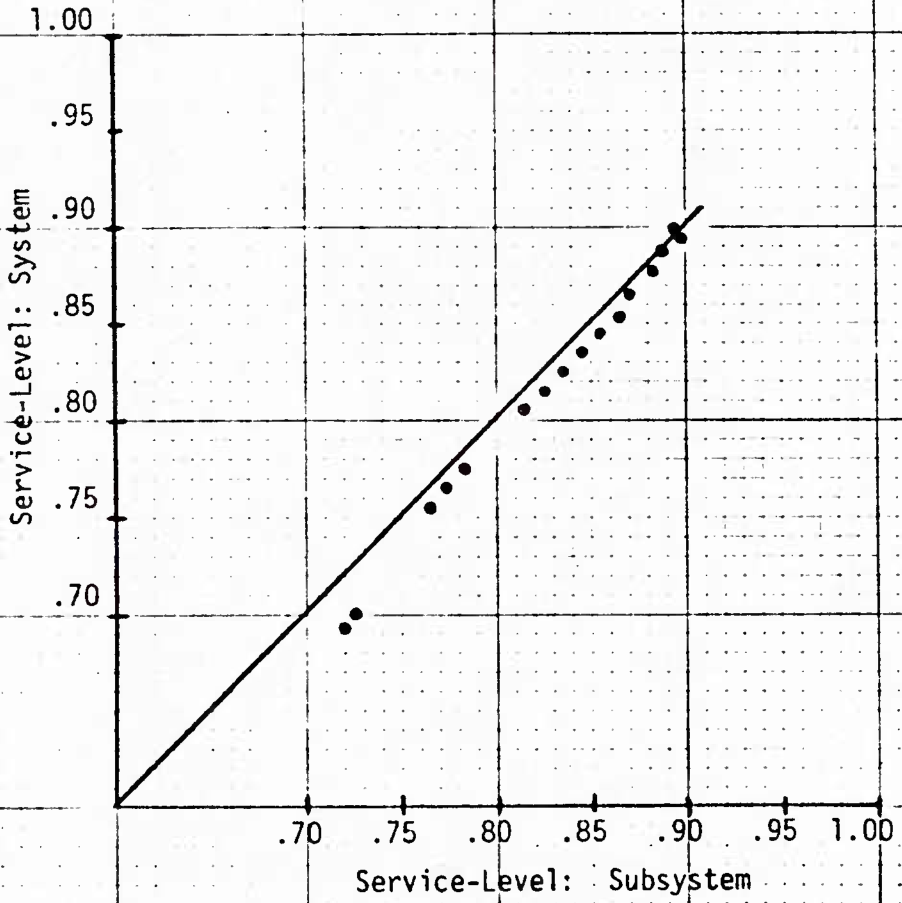


Figure 30. 128-Item Base-Case with Central Random Stratified Uniform Sampling Scheme 4

$S_i \geq D_i$

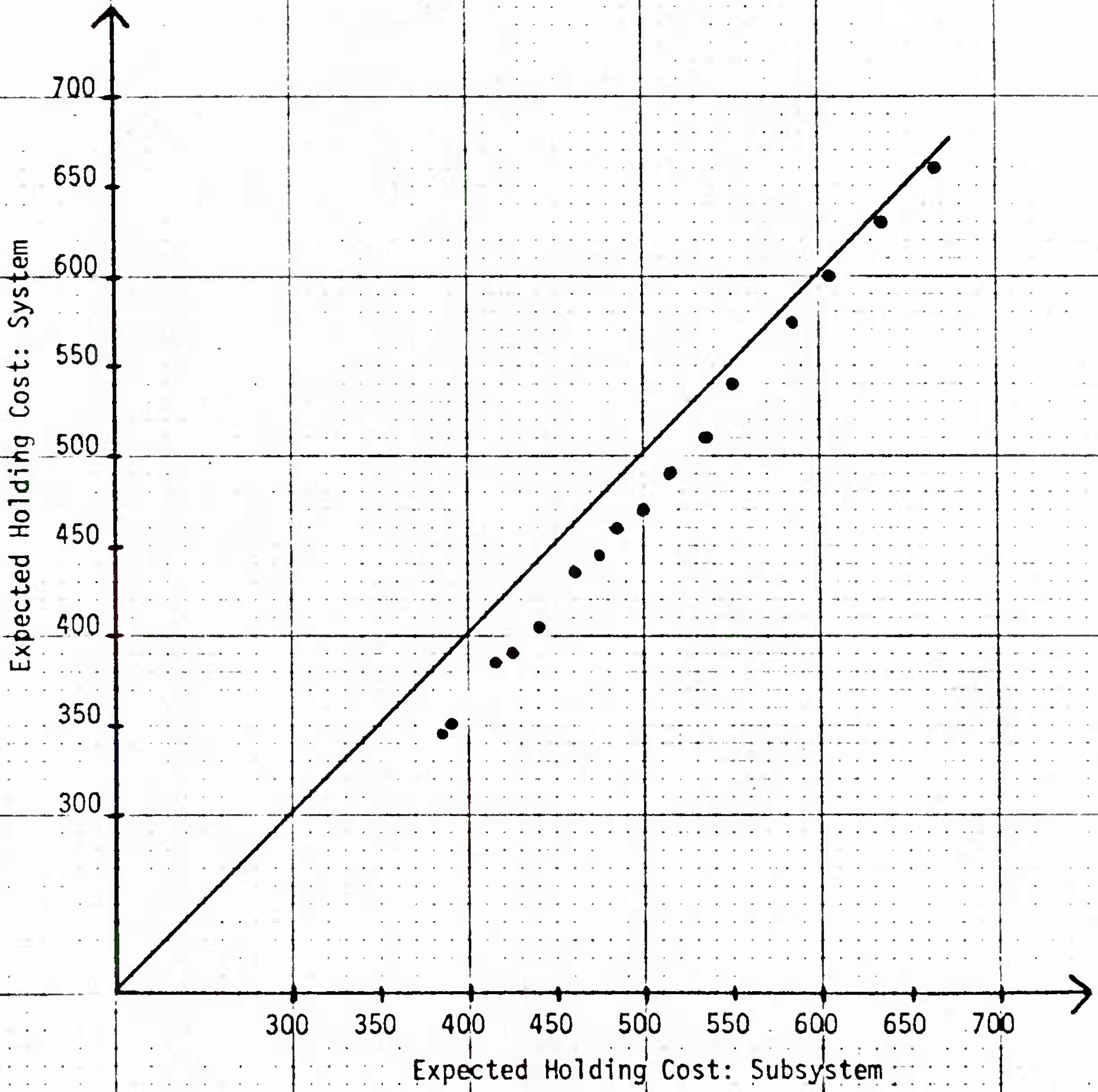


Figure 31. 128-Item Base-Case with Central Random Stratified Uniform Sampling Scheme 4

$S_i \geq D_i$

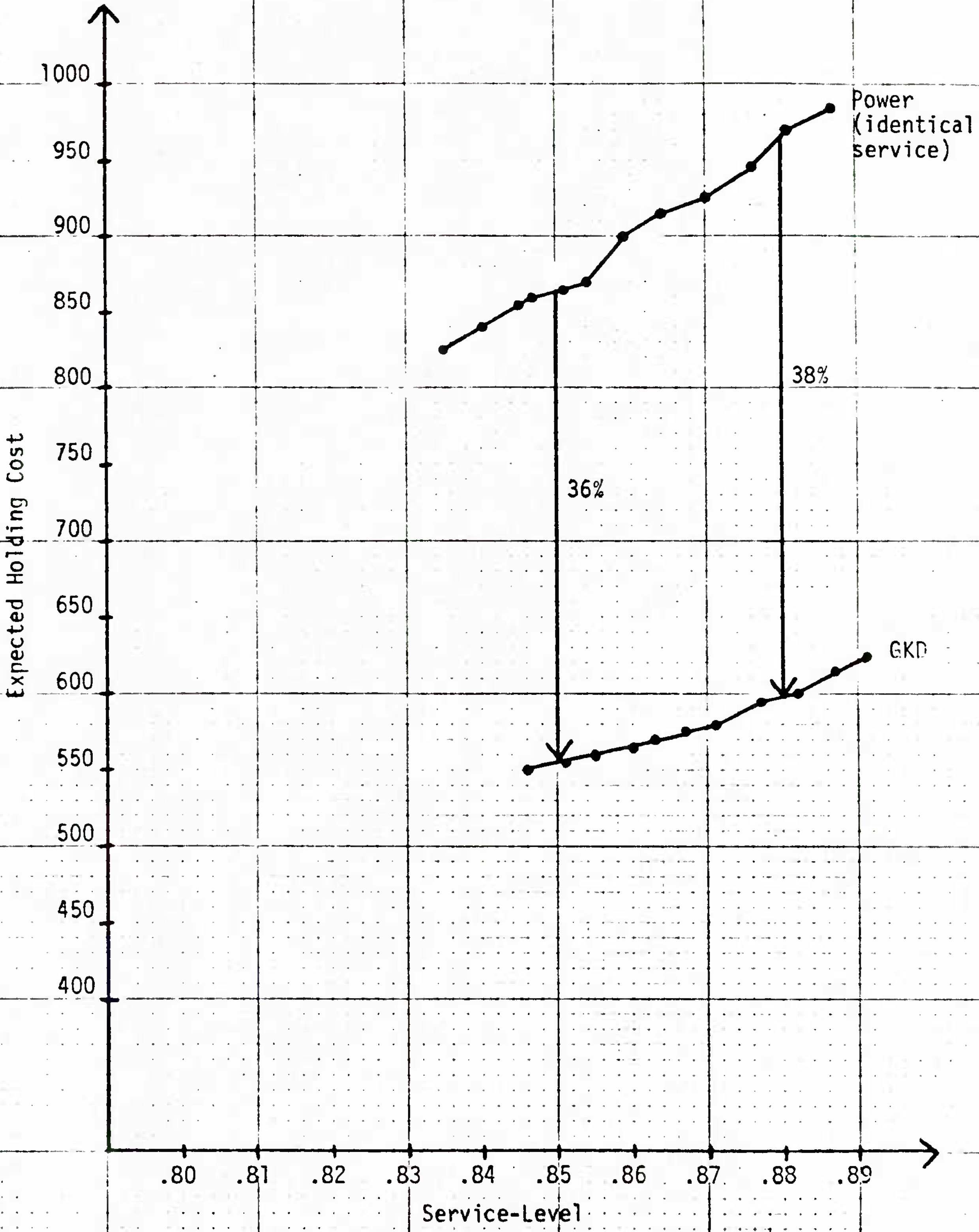


Figure 32. 128-Item Base-Case

$S_i \geq D_i + .75\mu_i^*$

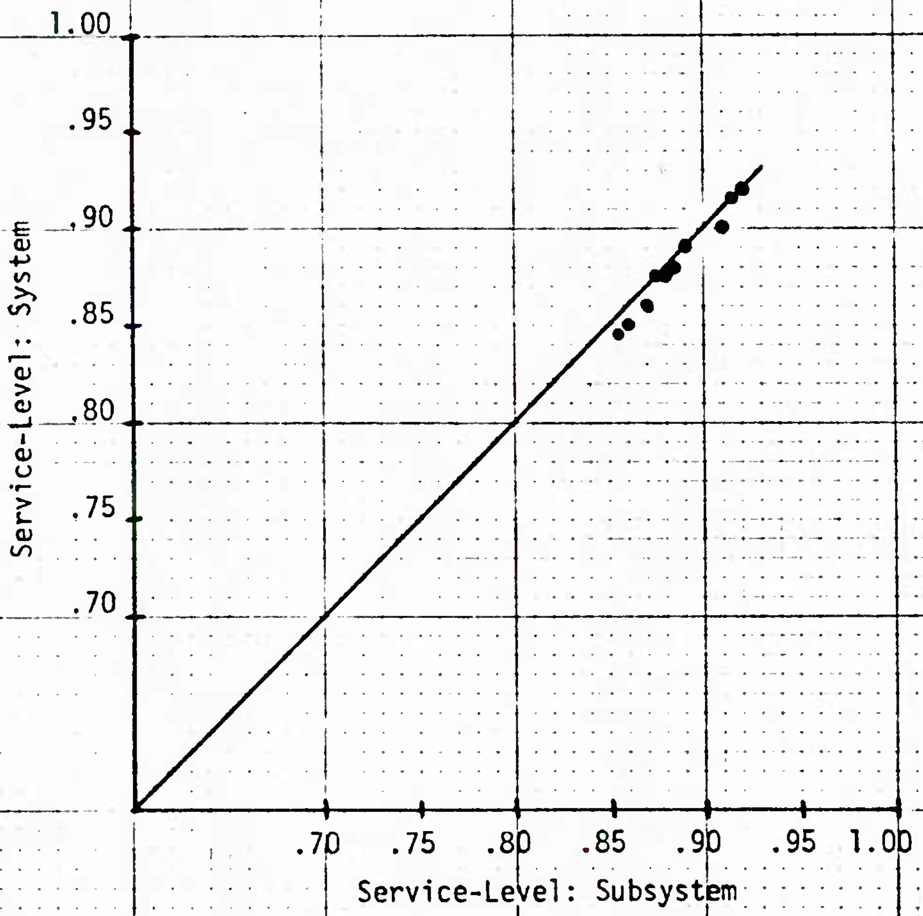


Figure 33. 128-Item Base Case with Central Random Stratified Uniform Sampling Scheme 4

$S_i \geq D_i + .75\mu_i$*

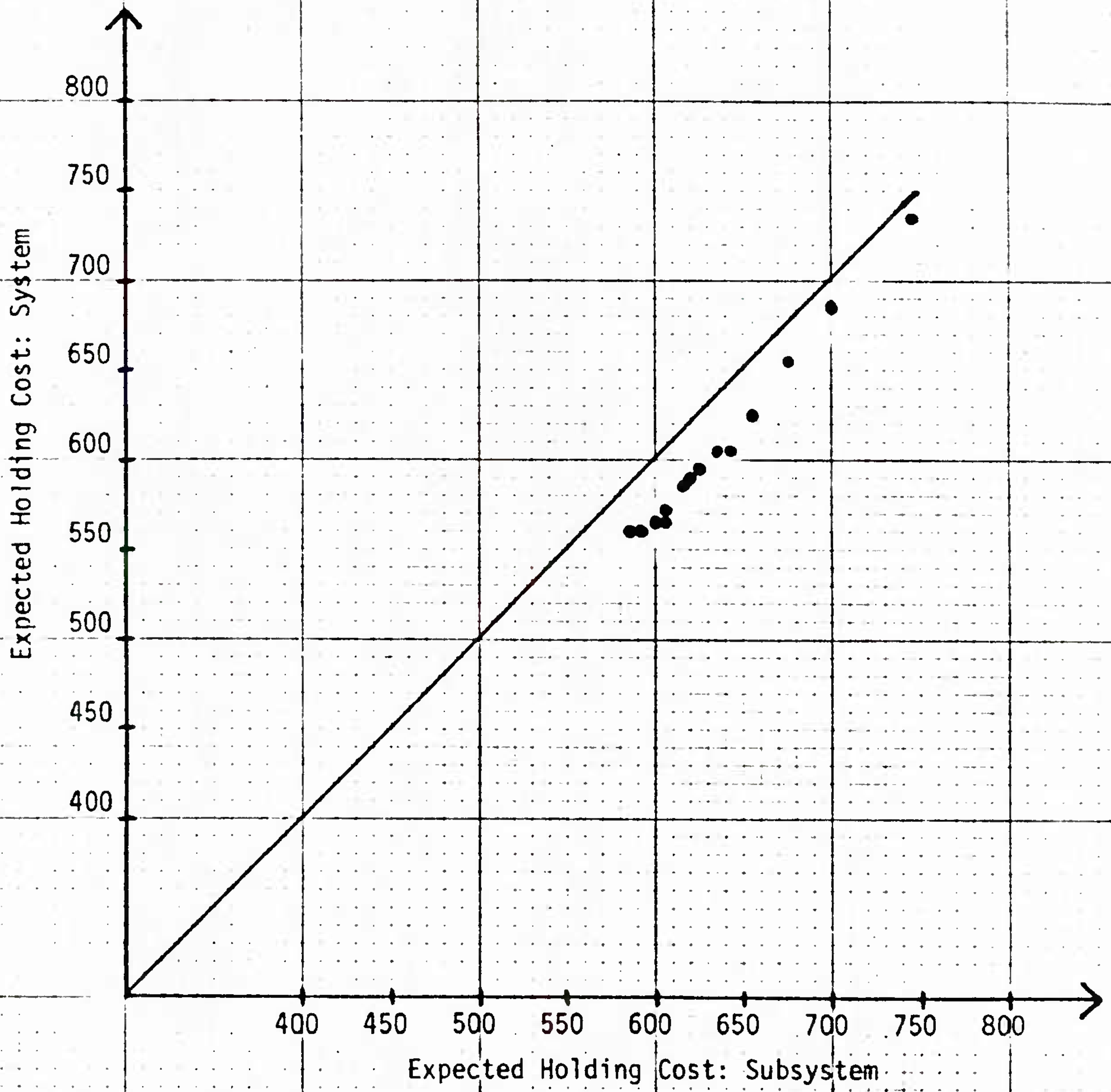


Figure 34 128-Item Base Case with Central Random Stratified Uniform Sampling Scheme 4 $S_i \geq D_i + .75\mu_i$*

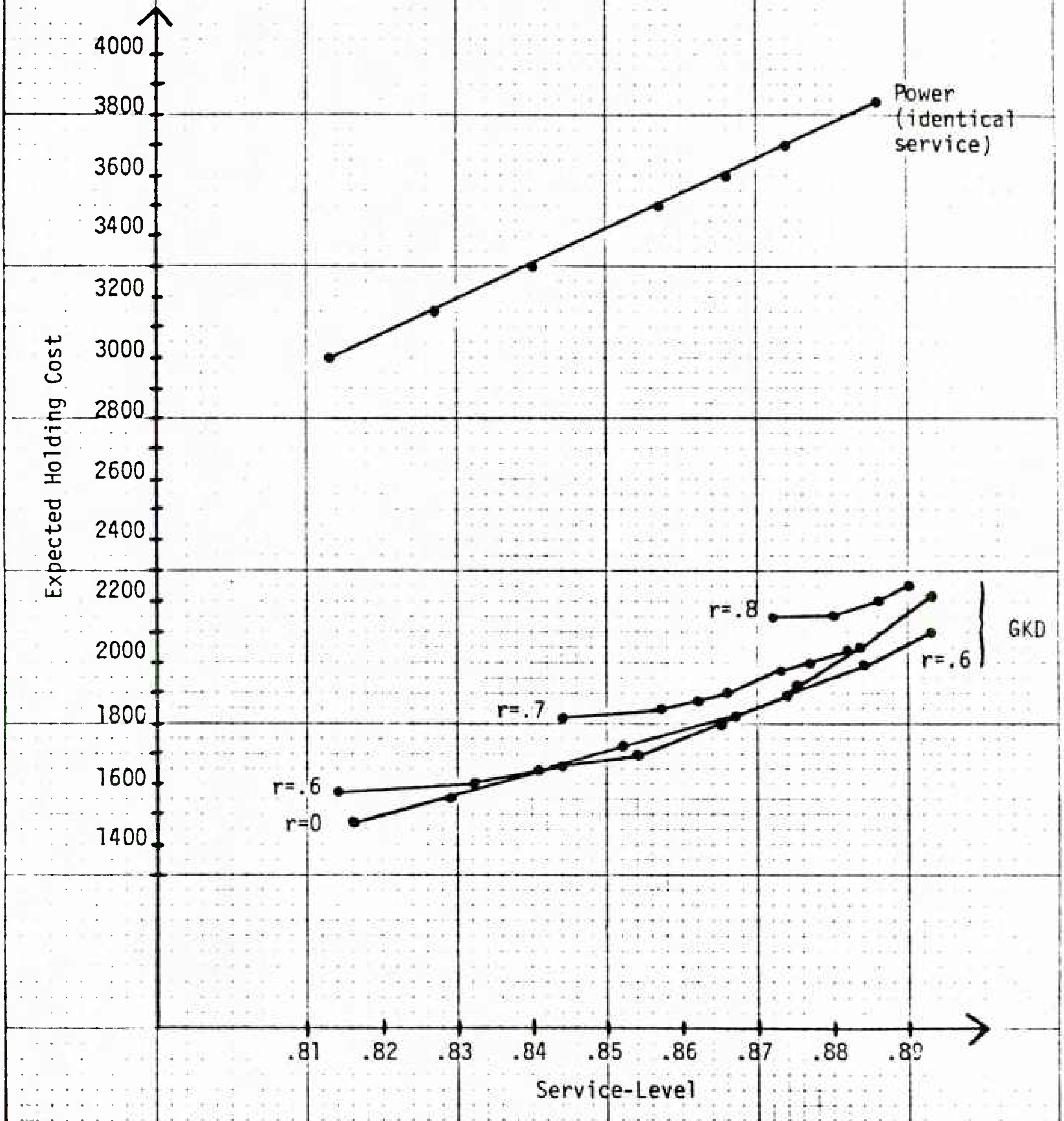


Figure 35. 128-Item Central Nonrandom Uniform Sampling from 512-Item

Base-Case System

$S_i \geq D_i + r\mu_i{}^*$

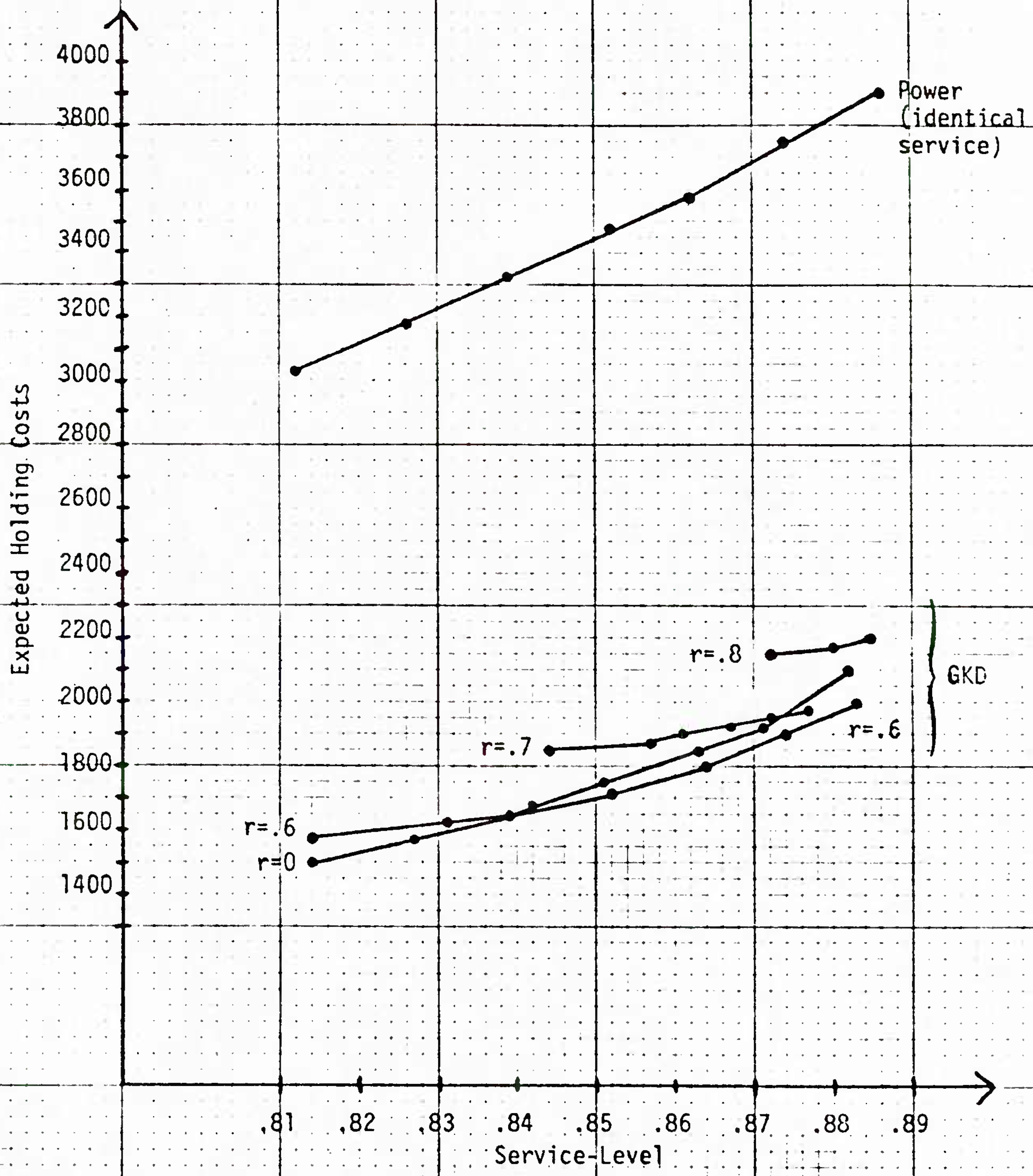


Figure 36. 512-Item Base-Case System

$S_i \geq D_i + r\mu_i^*$

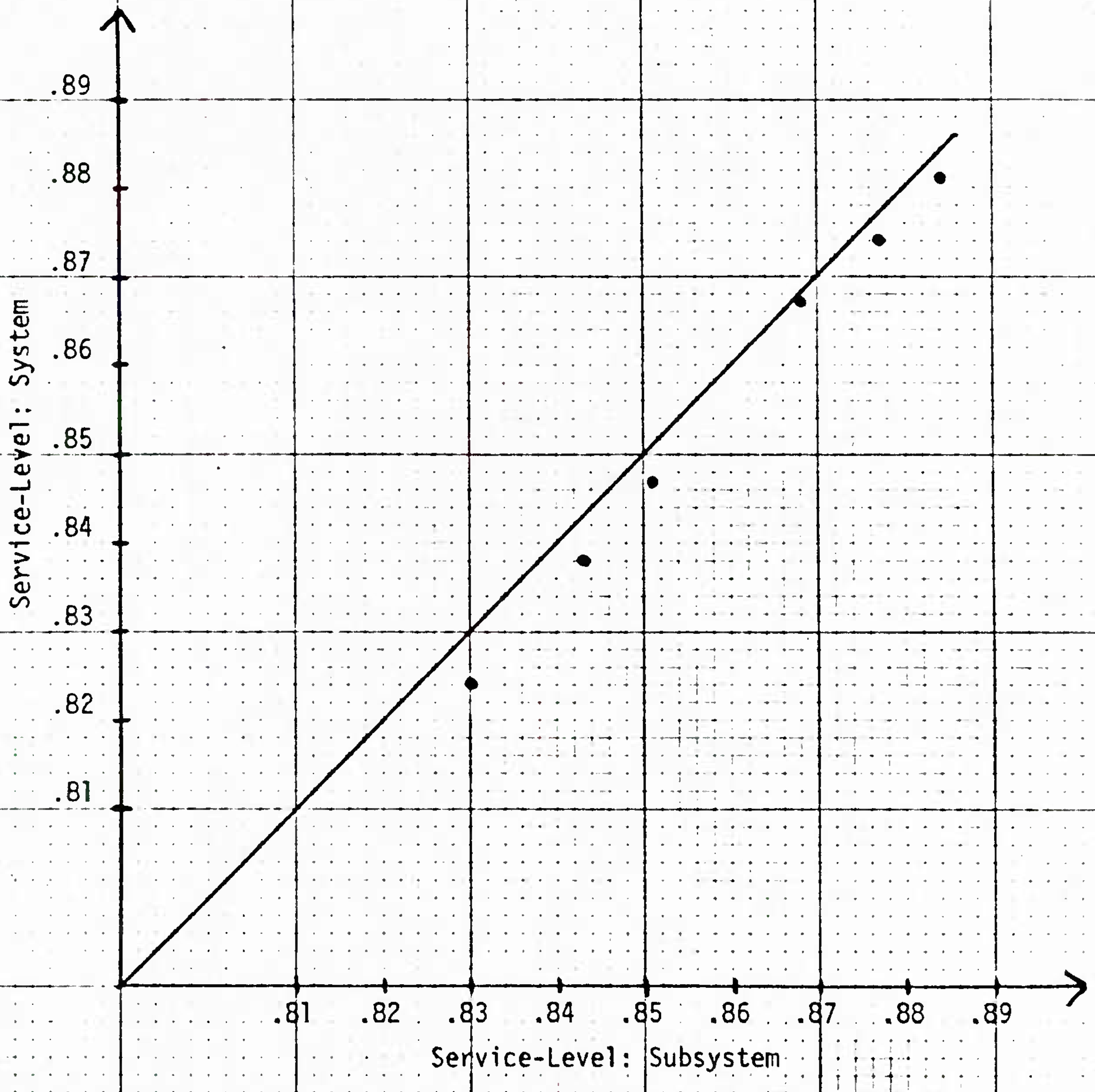


Figure 37. 512-Item Base-Case with Central Nonrandom Uniform Sampling Scheme (32-Item Sample)
$S_i \geq D_i + .6\psi_i^*$

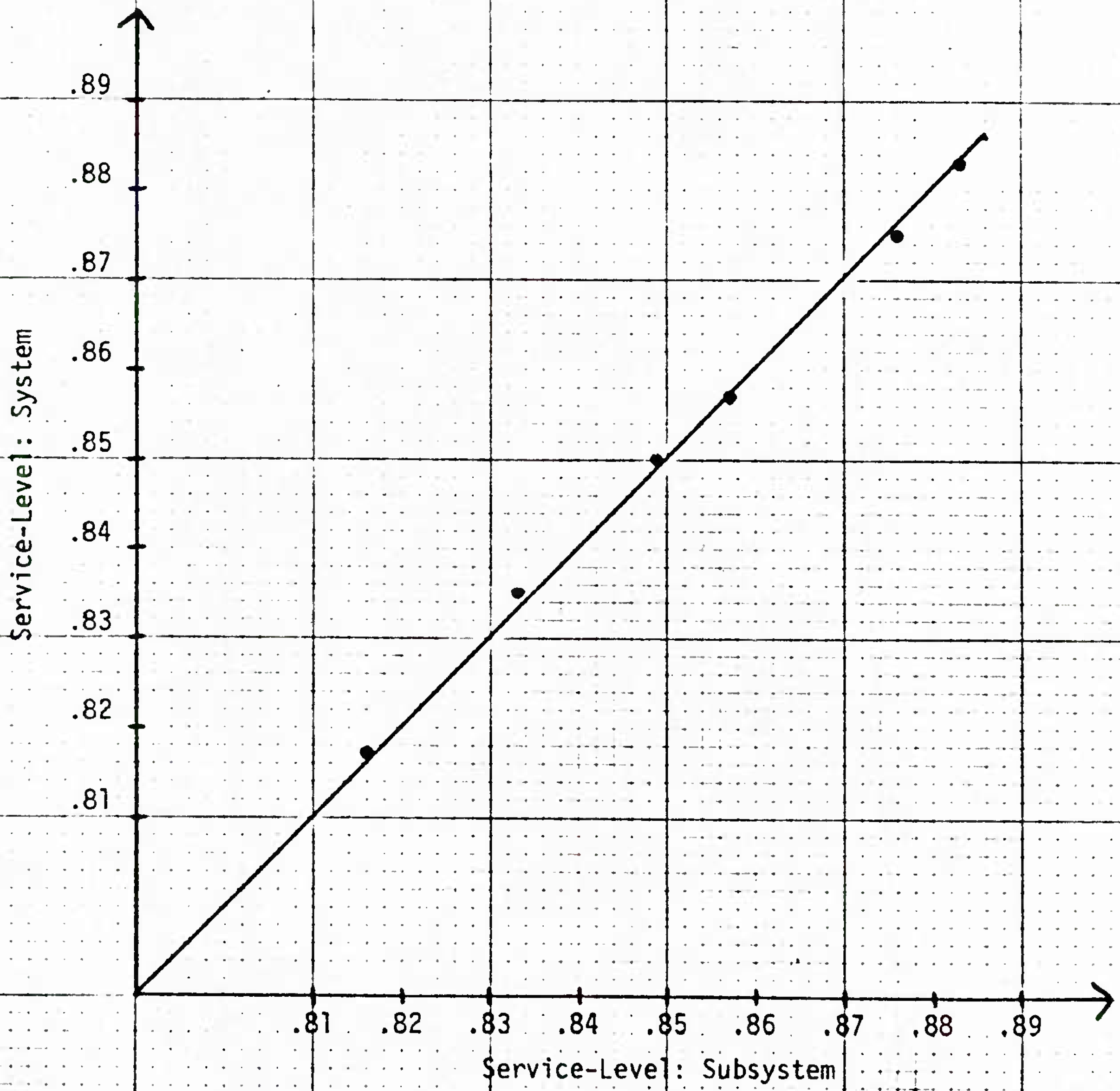


Figure 38. 512-Item Base-Case with Central Nonrandom Uniform Sampling Scheme (64-Item Sample) $S_i \geq D_i + .6\mu_i$*

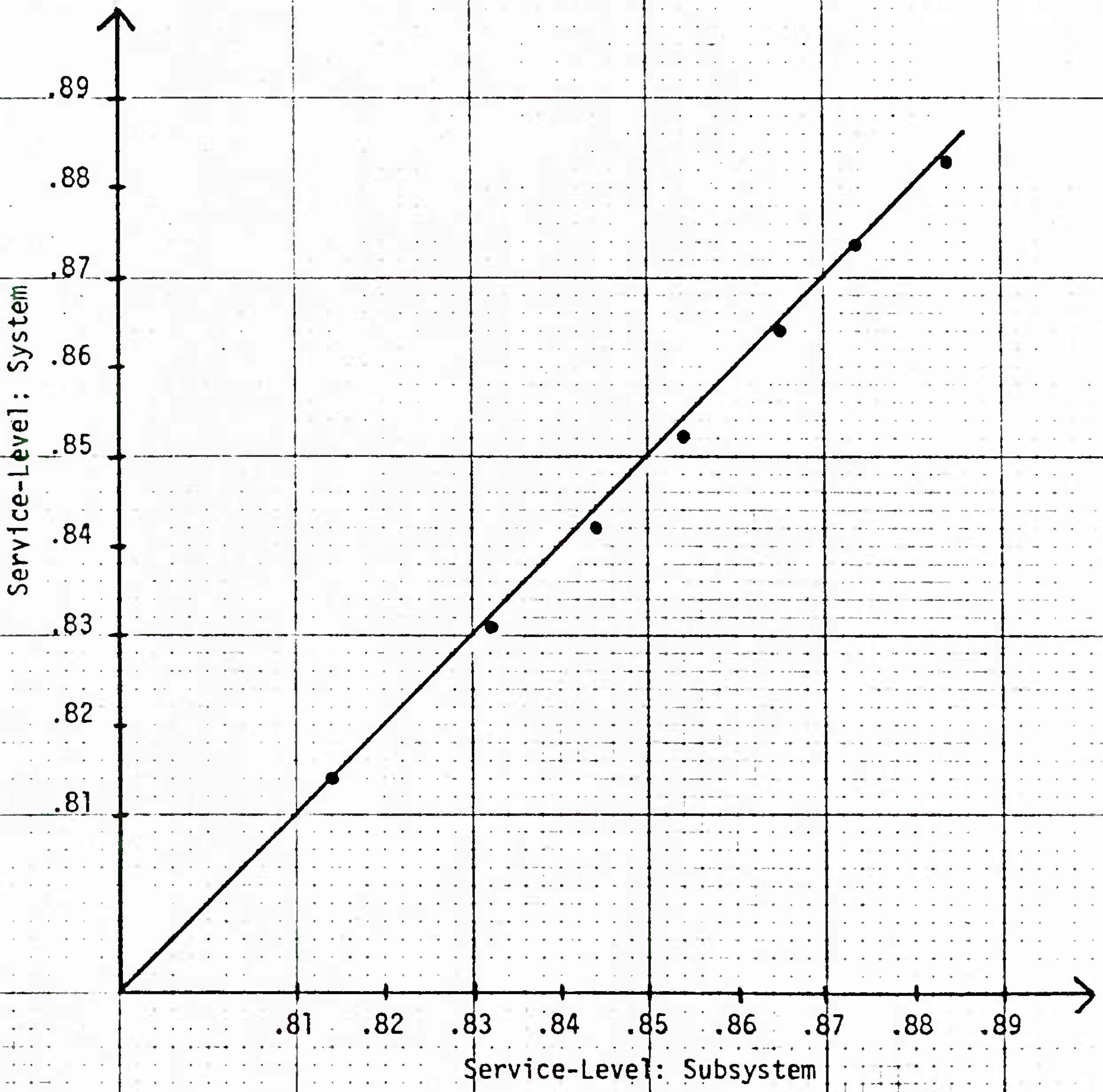


Figure 39. 512-Item Base-Case with Central Nonrandom Uniform Sampling Scheme (128-Item Sample)
$S_i \geq D_i + .6\mu_i^*$

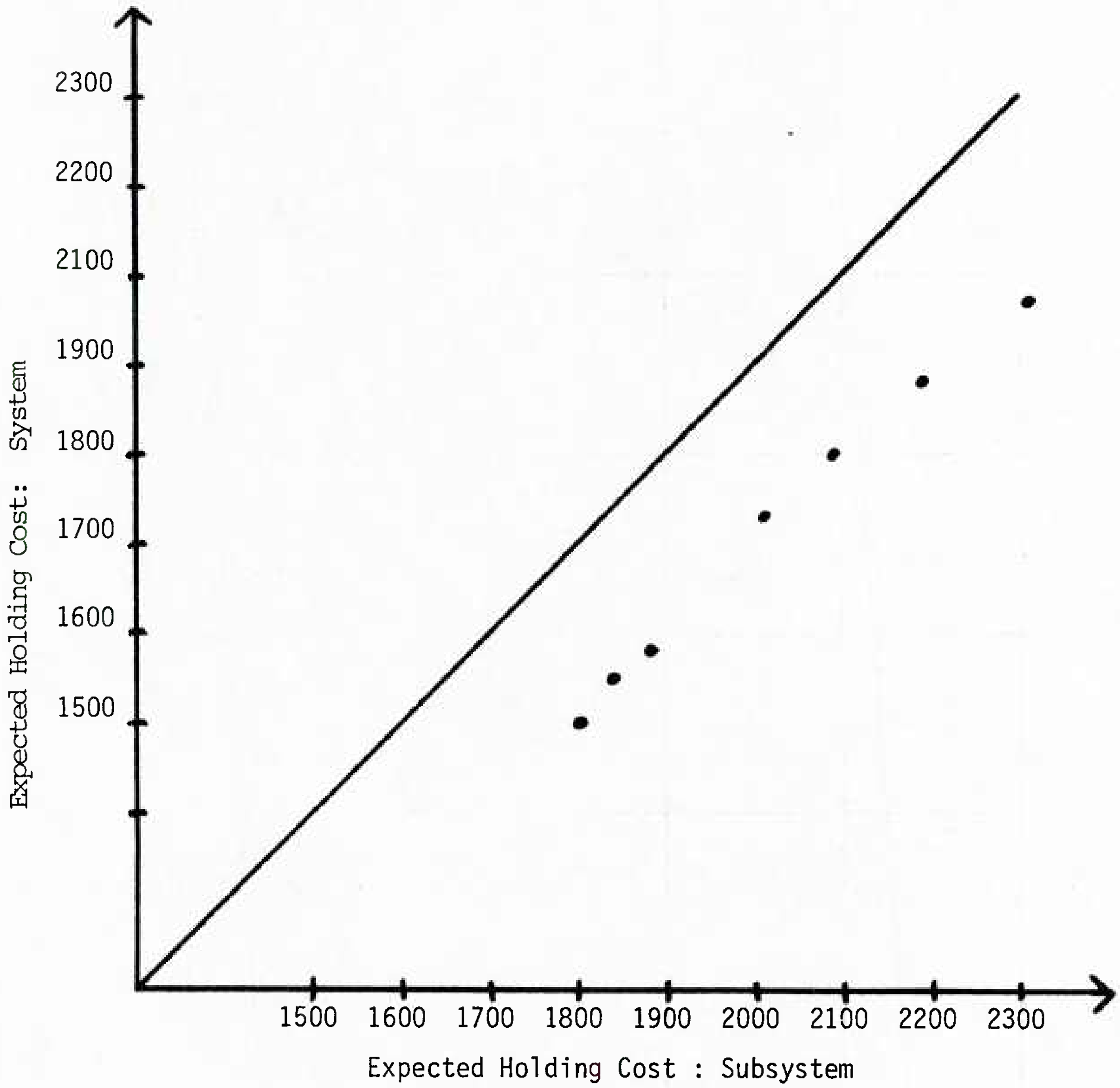


Figure 40. 512 - Item Base - Case with Central Nonrandom Uniform Sampling Scheme (32-Item Sample)

$S_i \geq D_i + .6\mu_i$*

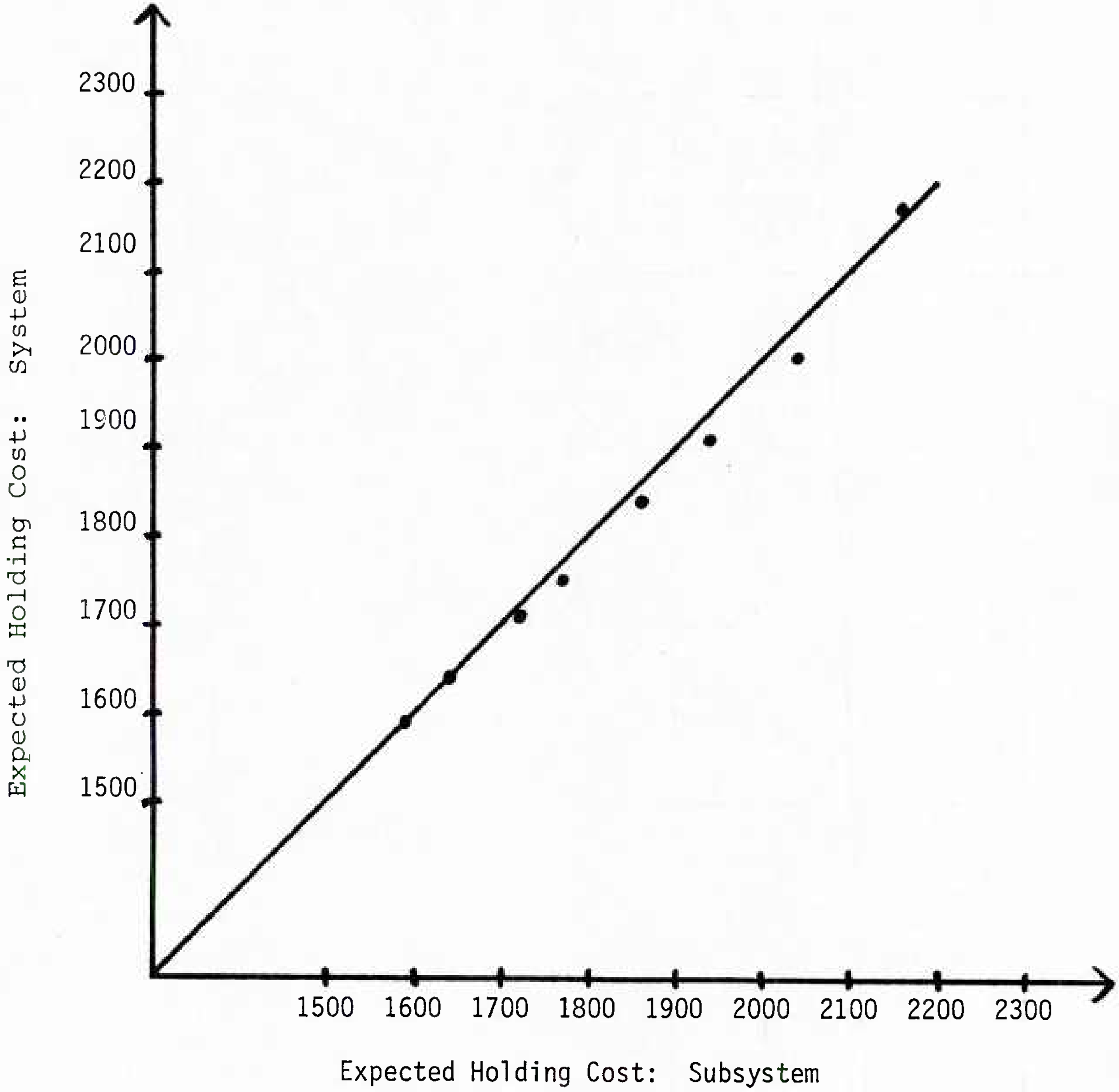


Figure 41. 512 - Item Base - Case with Central Nonrandom Uniform Sampling Scheme (64 - Item Sample)

$S_i \geq D_i + .6\ \mu_i$*

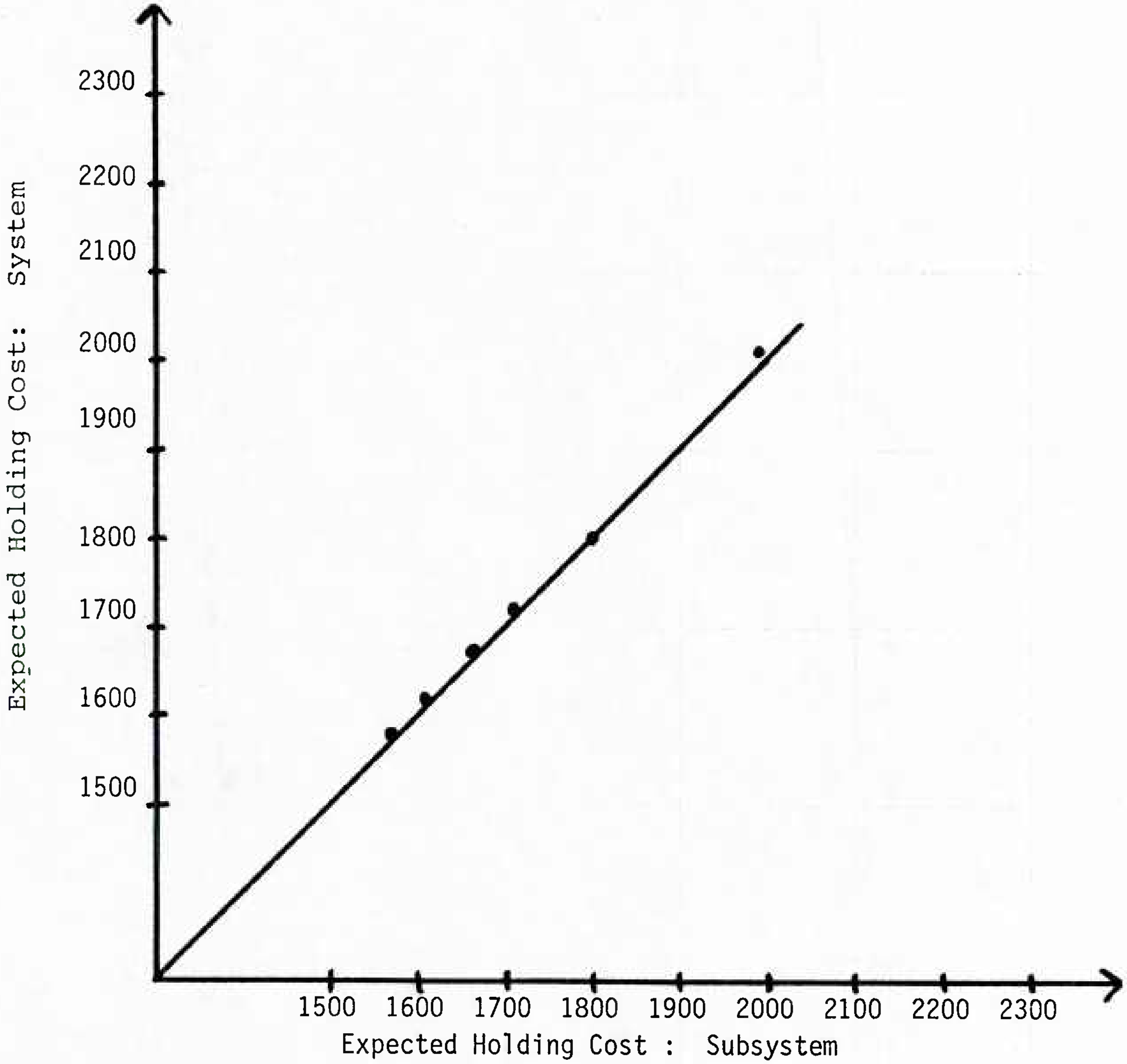


Figure 42. 512 - Item Base-Case with Central Nonrandom Uniform Sampling Scheme (128-Item Sample)

$S_i \geq D_i + .6\ \mu_i$*

APPENDIX TO CHAPTER IV

This appendix reports the seeds used in the IMSL Routine GGUD to create the six random samples of Scheme 4 of Section 4, and a listing of the 512-item inventory system of Section 5. The seed used in the IMSL Routine GGUD to generate the μ_i for this system is 872245376.

Table 14. Seeds for Six Tests of Central Random Stratified Uniform Sampling Scheme 4.

Test	Seed
1	96665107
2	227164307
3	929811759
4	2058317624
5	75813808
6	1743717576

NOTE: Experiment for Test 1 reported in Figures 29 and 30.

512-Item Inventory System of Chapter IV

For all items i = 1,2,...,512:

$$K_i = 24$$

$$k_i = 4$$

$$\frac{\sigma_i^2}{\mu_i} = 9$$

i	h_i	μ_i	σ_i	D_i	D_i/μ_i
1	1.8145	9.0000	9.0000	17.5010	1.9446
2	2.0363	8.0000	8.4853	15.7557	1.9695
3	2.7002	6.0000	7.3485	12.1972	2.0329
4	1.2363	13.0000	10.8167	24.6235	1.8941
5	2.6514	6.0000	7.3485	12.3104	2.0517
6	1.2090	13.0000	10.8167	24.9025	1.9156
7	0.9691	16.0000	12.0000	30.3035	1.8940
8	3.8186	4.0000	6.0000	8.7403	2.1851
9	1.0020	15.0000	11.6190	29.0211	1.9347
10	1.8469	8.0000	8.4853	16.5535	2.0692
11	1.1163	13.0000	10.8167	25.9282	1.9945
12	1.0957	13.0000	10.8167	26.1746	2.0134
13	1.0747	13.0000	10.8167	26.4318	2.0332
14	1.5218	9.0000	9.0000	19.1301	2.1256
15	0.8388	16.0000	12.0000	32.6002	2.0375
16	2.6290	5.0000	6.7082	11.5135	2.3027
17	0.9193	14.0000	11.2250	29.4759	2.1054
18	2.0997	6.0000	7.3485	13.8526	2.3088
19	1.5411	8.0000	8.4853	18.1413	2.2677
20	1.2063	10.0000	9.4868	22.4390	2.2439
21	0.9077	13.0000	10.8167	28.7900	2.2146
22	0.9618	12.0000	10.3923	27.0711	2.2559
23	2.2574	5.0000	6.7082	12.4364	2.4873
24	2.7593	4.0000	6.0000	10.3021	2.5755
25	1.0792	10.0000	9.4868	23.7387	2.3739

26	0.8793	12.0000	10.3923	28.3268	2.3606
27	0.8597	12.0000	10.3923	28.6517	2.3876
28	3.3623	3.0000	5.1962	8.3412	2.7804
29	1.4088	7.0000	7.9373	18.0101	2.5729
30	0.8035	12.0000	10.3923	29.6504	2.4709
31	4.7132	2.0000	4.2426	6.0171	3.0085
32	0.7680	12.0000	10.3923	30.3348	2.5279
33	3.0037	3.0000	5.1962	8.8310	2.9437
34	0.8811	10.0000	9.4868	26.3044	2.6304
35	2.8719	3.0000	5.1962	9.0338	3.0113
36	0.5266	16.0000	12.0000	41.2601	2.5788
37	0.9155	9.0000	9.0000	24.7401	2.7489
38	0.6715	12.0000	10.3923	32.4684	2.7057
39	0.4926	16.0000	12.0000	42.6775	2.6673
40	1.9272	4.0000	6.0000	12.3539	3.0885
41	0.6855	11.0000	9.9499	31.0273	2.8207
42	0.5269	14.0000	11.2250	39.0649	2.7904
43	0.4511	16.0000	12.0000	44.6212	2.7888
44	0.7846	9.0000	9.0000	26.7486	2.9721
45	1.7274	4.0000	6.0000	13.0572	3.2643
46	1.3523	5.0000	6.7082	16.1173	3.2235
47	2.2058	3.0000	5.1962	10.3242	3.4414
48	6.4770	1.0000	3.0000	3.9324	3.9324
49	0.7925	8.0000	8.4853	25.3993	3.1749
50	0.6206	10.0000	9.4868	31.4075	3.1407
51	0.5064	12.0000	10.3923	37.4529	3.1211
52	0.3718	16.0000	12.0000	49.2023	3.0751
53	1.1651	5.0000	6.7082	17.3793	3.4759
54	1.1410	5.0000	6.7082	17.5642	3.5128
55	0.3492	16.0000	12.0000	50.7896	3.1744
56	0.4975	11.0000	9.9499	36.4898	3.3173
57	0.4874	11.0000	9.9499	36.8728	3.3521
58	1.7507	3.0000	5.1962	11.6048	3.8683
59	0.6432	8.0000	8.4853	28.2280	3.5285
60	0.4584	11.0000	9.9499	38.0353	3.4578
61	0.4492	11.0000	9.9499	38.4273	3.4934
62	4.8424	1.0000	3.0000	4.5559	4.5559
63	0.2966	16.0000	12.0000	55.1623	3.4476
64	0.3877	12.0000	10.3923	42.8717	3.5726
65	0.2850	16.0000	12.0000	56.2865	3.5179
66	0.7452	6.0000	7.3485	23.3980	3.8997
67	0.3372	13.0000	10.8167	47.5148	3.6550
68	4.2988	1.0000	3.0000	4.8388	4.8388
69	0.4216	10.0000	9.4868	38.1969	3.8197
70	0.2953	14.0000	11.2250	52.3615	3.7401
71	1.3518	3.0000	5.1962	13.2271	4.4090
72	0.6630	6.0000	7.3485	24.8234	4.1372
73	1.9513	2.0000	4.2426	9.4012	4.7006
74	0.7658	5.0000	6.7082	21.4917	4.2983
75	0.3757	10.0000	9.4868	40.4908	4.0491

76	0.9216	4.0000	6.0000	17.9437	4.4859
77	0.3289	11.0000	9.9499	44.9928	4.0903
78	0.8876	4.0000	6.0000	18.2878	4.5719
79	0.2178	16.0000	12.0000	64.4933	4.0308
80	0.4276	8.0000	8.4853	34.7056	4.3382
81	0.5597	6.0000	7.3485	27.0443	4.5074
82	1.0990	3.0000	5.1962	14.6877	4.8959
83	1.6187	2.0000	4.2426	10.3336	5.1668
84	3.1788	1.0000	3.0000	5.6372	5.6372
85	0.2081	15.0000	11.6190	64.2834	4.2856
86	0.7665	4.0000	6.0000	19.6980	4.9245
87	0.6022	5.0000	6.7082	24.2694	4.8539
88	0.4930	6.0000	7.3485	28.8395	4.8066
89	0.3632	8.0000	8.4853	37.6953	4.7119
90	0.2854	10.0000	9.4868	46.5287	4.6529
91	0.4006	7.0000	7.9373	34.0286	4.8612
92	0.2296	12.0000	10.3923	55.8810	4.6567
93	2.7077	1.0000	3.0000	6.1138	6.1138
94	0.3801	7.0000	7.9373	34.9453	4.9922
95	2.6151	1.0000	3.0000	6.2224	6.2224
96	0.1606	16.0000	12.0000	75.2350	4.7022
97	0.2297	11.0000	9.9499	53.9564	4.9051
98	1.2417	2.0000	4.2426	11.8168	5.9084
99	0.3488	7.0000	7.9373	36.5007	5.2144
100	0.2182	11.0000	9.9499	55.3723	5.0338
101	0.5900	4.0000	6.0000	22.4861	5.6215
102	2.3206	1.0000	3.0000	6.6102	6.6102
103	0.1426	16.0000	12.0000	79.9024	4.9939
104	0.7481	3.0000	5.1962	17.8439	5.9480
105	0.5518	4.0000	6.0000	23.2612	5.8153
106	0.3101	7.0000	7.9373	38.7357	5.5337
107	2.1354	1.0000	3.0000	6.8944	6.8944
108	0.3001	7.0000	7.9373	39.3867	5.6267
109	0.1879	11.0000	9.9499	59.7316	5.4301
110	0.1452	14.0000	11.2250	74.9896	5.3564
111	0.4001	5.0000	6.7082	29.8499	5.9700
112	0.1789	11.0000	9.9499	61.2218	5.5656
113	0.4842	4.0000	6.0000	24.8512	6.2128
114	0.3177	6.0000	7.3485	36.0207	6.0035
115	0.9379	2.0000	4.2426	13.6199	6.8099
116	0.2637	7.0000	7.9373	42.0454	6.0065
117	0.1398	13.0000	10.8167	74.1927	5.7071
118	1.7887	1.0000	3.0000	7.5410	7.5410
119	0.1956	9.0000	9.0000	54.0145	6.0016
120	0.8667	2.0000	4.2426	14.1748	7.0874
121	0.2438	7.0000	7.9373	43.7518	6.2503
122	0.4201	4.0000	6.0000	26.7037	6.6759
123	0.8272	2.0000	4.2426	14.5132	7.2566
124	0.4073	4.0000	6.0000	27.1245	6.7811
125	0.2292	7.0000	7.9373	45.1417	6.4488

126	0.1436	11.0000	9.9499	68.4225	6.2202
127	0.1729	9.0000	9.0000	57.5011	6.3890
128	0.7663	2.0000	4.2426	15.0862	7.5431
129	1.5096	1.0000	3.0000	8.2170	8.2170
130	0.1859	8.0000	8.4853	52.9044	6.6130
131	0.1465	10.0000	9.4868	65.2118	6.5212
132	0.1203	12.0000	10.3923	77.5194	6.4599
133	0.1580	9.0000	9.0000	60.1889	6.6877
134	0.2001	7.0000	7.9373	48.3499	6.9071
135	0.2760	5.0000	6.7082	36.0159	7.2032
136	0.1700	8.0000	8.4853	55.3493	6.9187
137	0.1031	13.0000	10.8167	86.5519	6.6578
138	0.0825	16.0000	12.0000	105.3780	6.5861
139	0.0868	15.0000	11.6190	100.0795	6.6720
140	1.2828	1.0000	3.0000	8.9225	8.9225
141	0.1580	8.0000	8.4853	57.4299	7.1787
142	1.2462	1.0000	3.0000	9.0540	9.0540
143	0.0945	13.0000	10.8167	90.4492	6.9576
144	1.2109	1.0000	3.0000	9.1865	9.1865
145	0.3979	3.0000	5.1962	24.5588	8.1863
146	0.1681	7.0000	7.9373	52.8028	7.5433
147	0.0829	14.0000	11.2250	99.5967	7.1141
148	0.3814	3.0000	5.1962	25.0930	8.3643
149	0.2256	5.0000	6.7082	39.8857	7.9771
150	0.0856	13.0000	10.8167	95.1075	7.3160
151	0.0686	16.0000	12.0000	115.7561	7.2348
152	0.1545	7.0000	7.9373	55.1047	7.8721
153	0.0821	13.0000	10.8167	97.1407	7.4724
154	0.1315	8.0000	8.4853	63.0241	7.8780
155	0.0741	14.0000	11.2250	105.3869	7.5276
156	0.0640	16.0000	12.0000	119.8814	7.4926
157	0.0918	11.0000	9.9499	85.8229	7.8021
158	0.0905	11.0000	9.9499	86.4180	7.8562
159	0.0893	11.0000	9.9499	87.0152	7.9105
160	0.2423	4.0000	6.0000	35.2760	8.8190
161	0.0869	11.0000	9.9499	88.2160	8.0196
162	0.2359	4.0000	6.0000	35.7612	8.9403
163	0.1551	6.0000	7.3485	51.7659	8.6276
164	0.0918	10.0000	9.4868	82.5855	8.2586
165	0.0697	13.0000	10.8167	105.4965	8.1151
166	0.2981	3.0000	5.1962	28.4234	9.4745
167	0.1261	7.0000	7.9373	61.0821	8.7260
168	0.0968	9.0000	9.0000	77.1265	8.5696
169	0.0661	13.0000	10.8167	108.3613	8.3355
170	0.0530	16.0000	12.0000	131.8308	8.2394
171	0.0523	16.0000	12.0000	132.7069	8.2942
172	0.1033	8.0000	8.4853	71.2145	8.9018
173	0.0906	9.0000	9.0000	79.7237	8.8582
174	0.0503	16.0000	12.0000	135.3535	8.4596
175	0.3975	2.0000	4.2426	21.0302	10.5151

176	0.0872	9.0000	9.0000	81.3036	9.0337
177	0.1937	4.0000	6.0000	39.5102	9.8775
178	0.0588	13.0000	10.8167	114.9538	8.8426
179	0.2517	3.0000	5.1962	30.9631	10.3210
180	0.1864	4.0000	6.0000	40.2834	10.0709
181	0.7362	1.0000	3.0000	11.8171	11.8171
182	0.1817	4.0000	6.0000	40.8033	10.2008
183	0.0897	8.0000	8.4853	76.4770	9.5596
184	0.0709	10.0000	9.4868	94.1518	9.4152
185	0.1167	6.0000	7.3485	59.7951	9.9659
186	0.0532	13.0000	10.8167	120.9852	9.3066
187	0.2276	3.0000	5.1962	32.5827	10.8609
188	0.0674	10.0000	9.4868	96.5606	9.6561
189	0.0476	14.0000	11.2250	131.8974	9.4212
190	0.1316	5.0000	6.7082	52.3992	10.4798
191	0.0406	16.0000	12.0000	150.8694	9.4293
192	0.0642	10.0000	9.4868	99.0014	9.9001
193	0.3170	2.0000	4.2426	23.5789	11.7895
194	0.3132	2.0000	4.2426	23.7250	11.8625
195	0.0413	15.0000	11.6190	145.7934	9.7196
196	0.0470	13.0000	10.8167	128.7533	9.9041
197	0.0755	8.0000	8.4853	83.4607	10.4326
198	0.1193	5.0000	6.7082	55.0499	11.0100
199	0.0536	11.0000	9.9499	112.6709	10.2428
200	0.1165	5.0000	6.7082	55.7234	11.1447
201	0.0576	10.0000	9.4868	104.6114	10.4611
202	0.0948	6.0000	7.3485	66.4175	11.0696
203	0.0511	11.0000	9.9499	115.4288	10.4935
204	0.0505	11.0000	9.9499	116.1238	10.5567
205	0.0343	16.0000	12.0000	164.3162	10.2698
206	0.0904	6.0000	7.3485	68.0294	11.3382
207	0.0383	14.0000	11.2250	147.2022	10.5144
208	0.0408	13.0000	10.8167	138.4140	10.6472
209	0.0655	8.0000	8.4853	89.7047	11.2131
210	0.0324	16.0000	12.0000	169.2668	10.5792
211	0.0853	6.0000	7.3485	70.0733	11.6789
212	0.1686	3.0000	5.1962	37.9261	12.6420
213	0.0333	15.0000	11.6190	162.4096	10.8273
214	0.0353	14.0000	11.2250	153.3959	10.9568
215	0.0326	15.0000	11.6190	164.3150	10.9543
216	0.0805	6.0000	7.3485	72.1495	12.0249
217	0.0434	11.0000	9.9499	125.3620	11.3965
218	0.0674	7.0000	7.9373	83.8334	11.9762
219	0.4667	1.0000	3.0000	14.8825	14.8825
220	0.0288	16.0000	12.0000	179.4040	11.2127
221	0.4562	1.0000	3.0000	15.0543	15.0543
222	0.0322	14.0000	11.2250	160.6420	11.4744
223	0.0496	9.0000	9.0000	108.2049	12.0228
224	0.1103	4.0000	6.0000	52.5402	13.1351
225	0.0727	6.0000	7.3485	75.9687	12.6615

326	0.0145	11.0000	9.9499	218.2586	19.8417
327	0.0105	15.0000	11.6190	290.7210	19.3814
328	0.0131	12.0000	10.3923	238.3285	19.8607
329	0.0389	4.0000	6.0000	89.0698	22.2674
330	0.0193	8.0000	8.4853	166.6244	20.8280
331	0.0218	7.0000	7.9373	148.4071	21.2010
332	0.0504	3.0000	5.1962	69.8341	23.2780
333	0.0167	9.0000	9.0000	187.8061	20.8673
334	0.0744	2.0000	4.2426	49.1140	24.5570
335	0.0164	9.0000	9.0000	189.4807	21.0534
336	0.0365	4.0000	6.0000	91.8888	22.9722
337	0.0091	16.0000	12.0000	322.4315	20.1520
338	0.0160	9.0000	9.0000	192.0082	21.3342
339	0.0178	8.0000	8.4853	173.4085	21.6761
340	0.0101	14.0000	11.2250	289.2186	20.6585
341	0.0087	16.0000	12.0000	328.1514	20.5095
342	0.0116	12.0000	10.3923	253.5781	21.1315
343	0.0275	5.0000	6.7082	115.6913	23.1383
344	0.0682	2.0000	4.2426	51.3213	25.6606
345	0.0270	5.0000	6.7082	116.7033	23.3407
346	0.0191	7.0000	7.9373	158.5338	22.6477
347	0.0083	16.0000	12.0000	336.8383	21.0524
348	0.0329	4.0000	6.0000	96.8375	24.2094
349	0.0109	12.0000	10.3923	261.4045	21.7837
350	0.0216	6.0000	7.3485	140.4336	23.4056
351	0.0428	3.0000	5.1962	75.8883	25.2961
352	0.0085	15.0000	11.6190	324.4671	21.6311
353	0.0126	10.0000	9.4868	225.4156	22.5416
354	0.0313	4.0000	6.0000	99.3673	24.8418
355	0.0207	6.0000	7.3485	143.4757	23.9126
356	0.0123	10.0000	9.4868	228.3220	22.8322
357	0.0203	6.0000	7.3485	144.7030	24.1172
358	0.0086	14.0000	11.2250	312.5748	22.3268
359	0.0080	15.0000	11.6190	334.2946	22.2863
360	0.0132	9.0000	9.0000	211.1276	23.4586
361	0.0079	15.0000	11.6190	337.1332	22.4755
362	0.0146	8.0000	8.4853	191.4476	23.9310
363	0.0145	8.0000	8.4853	192.2550	24.0319
364	0.0384	3.0000	5.1962	80.1962	26.7321
365	0.0114	10.0000	9.4868	237.1690	23.7169
366	0.0087	13.0000	10.8167	302.1882	23.2452
367	0.0140	8.0000	8.4853	195.5042	24.4380
368	0.0074	15.0000	11.6190	347.1765	23.1451
369	0.0276	4.0000	6.0000	105.8552	26.4638
370	0.0073	15.0000	11.6190	350.0770	23.3385
371	0.0181	6.0000	7.3485	153.4614	25.5769
372	0.0098	11.0000	9.9499	266.2042	24.2004
373	0.0153	7.0000	7.9373	177.7266	25.3895
374	0.0177	6.0000	7.3485	155.3766	25.8961
375	0.0131	8.0000	8.4853	202.0963	25.2620

376	0.0075	14.0000	11.2250	336.9713	24.0694
377	0.0069	15.0000	11.6190	360.3380	24.0225
378	0.0103	10.0000	9.4868	250.2889	25.0289
379	0.1018	1.0000	3.0000	32.1526	32.1526
380	0.0084	12.0000	10.3923	297.7060	24.8088
381	0.0072	14.0000	11.2250	343.9351	24.5668
382	0.0083	12.0000	10.3923	300.1412	25.0118
383	0.0197	5.0000	6.7082	136.8934	27.3787
384	0.0082	12.0000	10.3923	302.5878	25.2156
385	0.0061	16.0000	12.0000	394.8866	24.6804
386	0.0963	1.0000	3.0000	33.0801	33.0801
387	0.0478	2.0000	4.2426	61.4488	30.7244
388	0.0316	3.0000	5.1962	88.5092	29.5031
389	0.0085	11.0000	9.9499	285.3096	25.9372
390	0.0117	8.0000	8.4853	214.7961	26.8495
391	0.0116	8.0000	8.4853	215.6586	26.9573
392	0.0131	7.0000	7.9373	191.9905	27.4272
393	0.0304	3.0000	5.1962	90.3006	30.1002
394	0.0056	16.0000	12.0000	409.4144	25.5884
395	0.0100	9.0000	9.0000	243.7023	27.0780
396	0.0074	12.0000	10.3923	317.5083	26.4590
397	0.0088	10.0000	9.4868	270.1981	27.0198
398	0.0097	9.0000	9.0000	246.6200	27.4022
399	0.0217	4.0000	6.0000	119.5422	29.8855
400	0.0108	8.0000	8.4853	223.5112	27.9389
401	0.0285	3.0000	5.1962	93.2100	31.0700
402	0.0065	13.0000	10.8167	349.6675	26.8975
403	0.0065	13.0000	10.8167	351.0436	27.0034
404	0.0167	5.0000	6.7082	148.8506	29.7701
405	0.0064	13.0000	10.8167	353.8052	27.2158
406	0.0206	4.0000	6.0000	122.8745	30.7186
407	0.0068	12.0000	10.3923	331.5509	27.6292
408	0.0135	6.0000	7.3485	178.0443	29.6741
409	0.0804	1.0000	3.0000	36.2354	36.2354
410	0.0160	5.0000	6.7082	152.3736	30.4747
411	0.0079	10.0000	9.4868	285.4337	28.5434
412	0.0112	7.0000	7.9373	207.6954	29.6708
413	0.0156	5.0000	6.7082	154.1530	30.8306
414	0.0387	2.0000	4.2426	68.3488	34.1744
415	0.0064	12.0000	10.3923	341.9855	28.4988
416	0.0085	9.0000	9.0000	264.5501	29.3945
417	0.0095	8.0000	8.4853	238.7892	29.8486
418	0.0375	2.0000	4.2426	69.4074	34.7037
419	0.0106	7.0000	7.9373	213.3618	30.4803
420	0.0739	1.0000	3.0000	37.8036	37.8036
421	0.0046	16.0000	12.0000	454.8307	28.4269
422	0.0091	8.0000	8.4853	243.3946	30.4243
423	0.0045	16.0000	12.0000	458.3055	28.6441
424	0.0055	13.0000	10.8167	380.6686	29.2822
425	0.0089	8.0000	8.4853	246.1825	30.7728

426	0.0044	16.0000	12.0000	463.5466	28.9717
427	0.0702	1.0000	3.0000	38.8216	38.8216
428	0.0696	1.0000	3.0000	38.9684	38.9684
429	0.0173	4.0000	6.0000	134.1998	33.5499
430	0.0049	14.0000	11.2250	416.5789	29.7556
431	0.0085	8.0000	8.4853	251.8137	31.4767
432	0.0085	8.0000	8.4853	252.7595	31.5949
433	0.0075	9.0000	9.0000	282.1588	31.3510
434	0.0061	11.0000	9.9499	339.6170	30.8743
435	0.0132	5.0000	6.7082	167.5711	33.5142
436	0.0073	9.0000	9.0000	285.3350	31.7039
437	0.0130	5.0000	6.7082	168.8233	33.7647
438	0.0065	10.0000	9.4868	316.1968	31.6197
439	0.0054	12.0000	10.3923	374.4240	31.2020
440	0.0046	14.0000	11.2250	432.4043	30.8860
441	0.0211	3.0000	5.1962	108.5672	36.1891
442	0.0314	2.0000	4.2426	75.9585	37.9793
443	0.0125	5.0000	6.7082	172.6129	34.5226
444	0.0062	10.0000	9.4868	323.2835	32.3284
445	0.0077	8.0000	8.4853	265.2435	33.1554
446	0.0153	4.0000	6.0000	142.9468	35.7367
447	0.0038	16.0000	12.0000	501.2124	31.3258
448	0.0120	5.0000	6.7082	175.8086	35.1617
449	0.0050	12.0000	10.3923	388.4489	32.3707
450	0.0037	16.0000	12.0000	506.7341	31.6709
451	0.0042	14.0000	11.2250	450.2121	32.1580
452	0.0053	11.0000	9.9499	362.8951	32.9905
453	0.0097	6.0000	7.3485	210.8340	35.1390
454	0.0041	14.0000	11.2250	455.1421	32.5101
455	0.0071	8.0000	8.4853	275.0872	34.3859
456	0.0063	9.0000	9.0000	307.0439	34.1160
457	0.0113	5.0000	6.7082	181.6476	36.3295
458	0.0080	7.0000	7.9373	246.5741	35.2249
459	0.0278	2.0000	4.2426	80.8092	40.4046
460	0.0050	11.0000	9.9499	373.5322	33.9575
461	0.0046	12.0000	10.3923	405.6792	33.8066
462	0.0060	9.0000	9.0000	313.7392	34.8599
463	0.0090	6.0000	7.3485	218.5636	36.4273
464	0.0077	7.0000	7.9373	251.9348	35.9907
465	0.0089	6.0000	7.3485	220.1292	36.6882
466	0.0035	15.0000	11.6190	506.0466	33.7364
467	0.0263	2.0000	4.2426	83.1531	41.5765
468	0.0065	8.0000	8.4853	288.2001	36.0250
469	0.0065	8.0000	8.4853	289.2236	36.1530
470	0.0086	6.0000	7.3485	224.0720	37.3453
471	0.0032	16.0000	12.0000	546.3857	34.1491
472	0.0507	1.0000	3.0000	45.7480	45.7480
473	0.0056	9.0000	9.0000	326.2352	36.2484
474	0.0250	2.0000	4.2426	85.2364	42.6182
475	0.0033	15.0000	11.6190	522.4059	34.8271

476	0.0070	7.0000	7.9373	262.8604	37.5515
477	0.0035	14.0000	11.2250	493.9922	35.2852
478	0.0243	2.0000	4.2426	86.4406	43.2203
479	0.0035	14.0000	11.2250	497.4593	35.5328
480	0.0060	8.0000	8.4853	300.6247	37.5781
481	0.0034	14.0000	11.2250	500.9408	35.7815
482	0.0032	15.0000	11.6190	535.3425	35.6895
483	0.0067	7.0000	7.9373	269.3603	38.4800
484	0.0029	16.0000	12.0000	571.8179	35.7386
485	0.0232	2.0000	4.2426	88.5719	44.2860
486	0.0461	1.0000	3.0000	48.0402	48.0402
487	0.0030	15.0000	11.6190	544.6977	36.3132
488	0.0076	6.0000	7.3485	238.6094	39.7682
489	0.0075	6.0000	7.3485	239.4329	39.9055
490	0.0034	13.0000	10.8167	483.0563	37.1582
491	0.0032	14.0000	11.2250	518.5639	37.0403
492	0.0147	3.0000	5.1962	130.1654	43.3885
493	0.0146	3.0000	5.1962	130.6121	43.5374
494	0.0029	15.0000	11.6190	557.9564	37.1971
495	0.0036	12.0000	10.3923	456.9109	38.0759
496	0.0054	8.0000	8.4853	317.6751	39.7094
497	0.0071	6.0000	7.3485	246.0816	41.0136
498	0.0047	9.0000	9.0000	355.7140	39.5238
499	0.0026	16.0000	12.0000	602.0170	37.6261
500	0.0419	1.0000	3.0000	50.3990	50.3990
501	0.0035	12.0000	10.3923	466.3276	38.8606
502	0.0034	12.0000	10.3923	467.9082	38.9923
503	0.0103	4.0000	6.0000	174.6764	43.6691
504	0.0031	13.0000	10.8167	506.6400	38.9723
505	0.0029	14.0000	11.2250	543.8452	38.8461
506	0.0067	6.0000	7.3485	253.6908	42.2818
507	0.0400	1.0000	3.0000	51.6036	51.6036
508	0.0031	13.0000	10.8167	513.5010	39.5001
509	0.0028	14.0000	11.2250	551.2000	39.3714
510	0.0098	4.0000	6.0000	178.8342	44.7085
511	0.0028	14.0000	11.2250	554.8994	39.6357
512	0.0129	3.0000	5.1962	139.2730	46.4243

V. CONCLUSION: IMPLEMENTATION OF THE GKD ALGORITHM WITH SAMPLING FOR A LARGE-SCALE INVENTORY SYSTEM

1. INTRODUCTION

This chapter concludes the studies in Chapter II. III and IV. in which we examined the problem of specifying single-item service objectives in a multi-item inventory system subject to an overall service objective. A list of the notation we use may be found at the end of Chapter II.

In that chapter we formulated this inventory problem as a constrained nonlinear program (NLP) and developed the Generalized Knapsack Duality (GKD) Algorithm to solve it. This algorithm finds the Lagrange constraint multiplier of the NLP and from it computes the (s,S) policies for all items in the inventory system. In Chapter III we reported computational experience with several 32-item inventory systems which have a structure typical of many real-world inventory systems. This structure is called an 80/20 value structure, and is one in which twenty percent of the items represent eighty percent of the value of the systems, where value is taken to be $h\mu$. These items are the high-value items, and the other eighty percent are the low-value items. In order to investigate the performance of the GKD Algorithm, we assumed that the demand distribution is Negative Binomial and compared the (exact) expected holding cost of the (s,S) policies produced by the GKD Algorithm with the (s,S) policies produced by a method popular with managers, the Identical Service Approach (The replenishment costs for these methods are the same, and represent about half of the total expected cost). Given a targeted overall service-level α (we targeted 85%), the Identical Service Approach sets (s,S) policies by the Power Approximation of Ehrhardt (Mosier [1981]) so that every item has a service-level α. (The GKD Algorithm varies individual service-levels while still maintaining an average overall service-level α.) We showed that there is a significant cost savings when using the GKD Algorithm rather than the Identical Service Approach to manage these inventory systems.

In Chapter IV we reported several sensitivity tests performed on the base-case and worst-case inventory systems of Chapter III. In particular, we performed sensitivity tests on the D_i and on the lower bounds for S_i. We also examined several sampling schemes and sizes for inventory aggregation for larger (128- and 512-item) systems. Since most real-world inventory systems contain thousands of items, typically managers sample from the system and make decisions about policies for all items in the system based on the sample. We examined sampling schemes (both random and nonrandom) of several sizes which maintained the 80/20 value structure and the overall service constraint weights. We summarize the conclusions and recommendations given in that chapter as follows:

1. Specify all D_i using Ehrhardt's Power Method (Mosier [1981]).
2. Specify the lower bounds on S_i as $D_i + r\mu_i^*$, where r is taken as large as possible while still including as feasible the targeted service-level and while the expected holding cost is significantly smaller than that of the Identical Service Approach.
3. Sample about 32 items using a central nonrandom uniform sampling scheme to create the inventory subsystem for aggregation (sample items from a stratum containing items 1,2,...,N by taking items N/2M, 3N/2M, 5N/2M,...,(2M-1)N/2M, where the fractions are rounded to the nearest integer). Recompute the holding and replenishment costs to maintain the value structure of the system, and recompute the constraint weights to maintain the constraint-weight structure of the system (multiply the costs by $\sum_{\substack{i \text{ in} \\ \text{system}}} h_i\mu_i / \sum_{\substack{i \text{ in} \\ \text{sample}}} h_i\mu_i$ and the constraint weights by N/M for each of ihe M items sampled from 1,2,...N).

In Section 2 of this chapter we prescribe a detailed implementation procedure of the GKD Algorithm with sampling for a large-scale inventory system with an overall service objective. In Section 3 we report the results of implementing this procedure for the 512-item system described in Chapter IV. Section 4 contains conclusions and directions for future research, and the Appendix contains a listing of the FORTRAN code of the GKD Algorithm and related programs.

2. IMPLEMENTATION PROCEDURE

Based on the conclusions and recommendations of Chapters II, III, and IV, we offer the following procedure to specify (s,S) policies in large-scale inventory systems. Our empirical evidence suggests that this procedure is a significant improvement over the Identical Service Approach often used by managers, capable of substantially reducing operating costs in an inventory system with an overall service objective.

The procedure is as follows:

1. Determine the desired range of service-levels and the targeted service-level. This is a managerial decision, but typically the range is 80% to 90% with a target of 85%.
2. Determine r as follows. For r = 0,.5,.6,.7,.8,.9 and 1, input the following values to the program LSMISS (see the Appendix to this chapter):

   TITLE = user-supplied title

   LBAL = smallest service-level to consider

   UBAL = largest service-level to consider

   INCRAL = service-level increment between LBAL and UBAL

   NSIG = 8

   MAXFN = 100

   NI = number of items in inventory system

   SCHEME = 1

   ENTIRE = 0

   STRA(1) = NI

   NUMB(1) = about 32.

   Using the output from LSMISS, graph the expected holding costs for the various service-levels for each value of r and for the

Identical Service Approach (compare Figure 34). Take as r the largest value of r such that LBAL is achievable and such that there is an acceptable cost savings below that of the Identical Service Approach.

NOTE 1:

The GKD Algorithm assumes that the demand distribution is Logistic, an approximation to the Normal distribution (Section 2 of Chapter II). If the actual distribution is not well-approximated by a Normal distribution, then the service-level of the (s,S) policies produced by the GKD Algorithm may be different from the specified service-level. The user may therefore have to try various LBAL, UBAL and INCRAL. If the service-level specifications are as described in Step (1) and the demand distribution is well-approximated by a Gamma or Negative Binomial distribution, our experience suggests using

LBAL = .75

UBAL = .85

INCRAL = .01.

NOTE 2:

We recommend a sample size (NUMB(1)) of about 32 because in our experience larger samples do not significantly improve the performance of the sampling scheme.

NOTE 3:

The parameters NSIG and MAXFN are used to terminate IMSL subroutines in the GKD Algorithm. We have found the recommended values to be satisfactory in every experiment performed in this paper.

3. Using the output from LSMISS and linear interpolation, determine λ^* for a targeted service-level. Input the following values to the program MISSNLSI (see the Appendix to this chapter):

TITLE = user-supplied title

LS = λ^*

NSIG = 8

MAXFN = 100

NI = number of items in system

$$\text{OUT} = \begin{cases} 0 & \text{write output only to printer} \\ 1 & \text{write output to both printer and an external file} \end{cases}$$

The output of MISSNLSI contains all single-item policy specifications.

NOTE:

The parameter OUT allows the user to save the output of MISSNLSI in an external file. See the Appendix to this chapter for details.

3. EXAMPLE

In the previous chapter we performed Steps (1) and (2) of this procedure for the 512-item base-case inventory system, and decided to use r=.6. We now perform Step (3) for this system. We used the output from LSMISS for the scheme described in Step (2) of the procedure, which corresponds to Experiment 1 of Table 3. For this experiment, Figure 43 reports the values of λ^* for the service-levels near 85%, our targeted service-level. Linear interpolation gives λ^* = 4207. The output from the program MISSNLSI is summarized in Figures 44, 45, and 46. The operating characteristics of the individual items follow the same pattern noted in Chapter III, in that the high-value items are stocked at low service-levels and the low-value items are stocked at high service-levels.

4. CONCLUSIONS, FUTURE RESEARCH DIRECTIONS

Using the GKD Algorithm of Chapter II and the extensive computational and sensitivity experience reported in Chapters III and IV, we have described a detailed implementation procedure of the GKD Algorithm using sampling to manage a large-scale inventory system with an overall service objective. We have shown that for inventory systems with a structure typical of that found in many real-world inventory systems, this procedure is practical and results in a significant cost savings below that of the Identical Service Approach often used by managers.

We recommend the following future research directions:

1. This procedure should be tested in a statistical environment in which the moments of the demand distribution are estimated from recent demand history. In light of the results reported in Ehrhardt [1976], Estey and Kaufman [1975], Klincewicz [1976a], MacCormick [1974, 1977], and MacCormick et al. [1977], the performance of the procedure will probably degrade somewhat, and this should be investigated. Considering the cost savings, however, we conjecture that in a statistical environment our procedure will still result in significant cost savings below that of the Identical Service Approach.

2. The variable r should be included in the GKD optimization routine. This would improve accuracy and make an implementation procedure simpler in that Step (2) of the procedure would be eliminated.
3. The GKD Algorithm assumes the demand distribution is Logistic, while demand distributions are typically skewed like a Gamma or Negative Binomial distribution. As reported in Section 3 of Chapter III, this can lead to significant degradation in algorithm performance (although we showed it is possible to improve such performance). Moreover, as discussed in Note 1 under Step (2) of our implementation procedure, the actual service-level of the policies produced by the GKD Algorithm may differ from that specified. We recommend that the Logistic distribution be replaced by a distribution which better approximates a Gamma (or Negative Binomial) distribution. The distribution to use is not obvious; an examination of the proof of Theorem 8 of Chapter II suggests that the distribution function must have a fairly simple form in order to prove algorithm convergence in the way we did ($\lambda(\theta)$ needs to be rational in θ). We have been unable to find a distribution that better approximates a Gamma distribution and yet is simple enough to allow an extension of the proof of Theorem 8.
4. Inventory systems with a value structure different than 80/20 should be studied. Based on previous but unreported experiments, we conjecture that the algorithm performance will degrade for X/20 systems as X decreases.

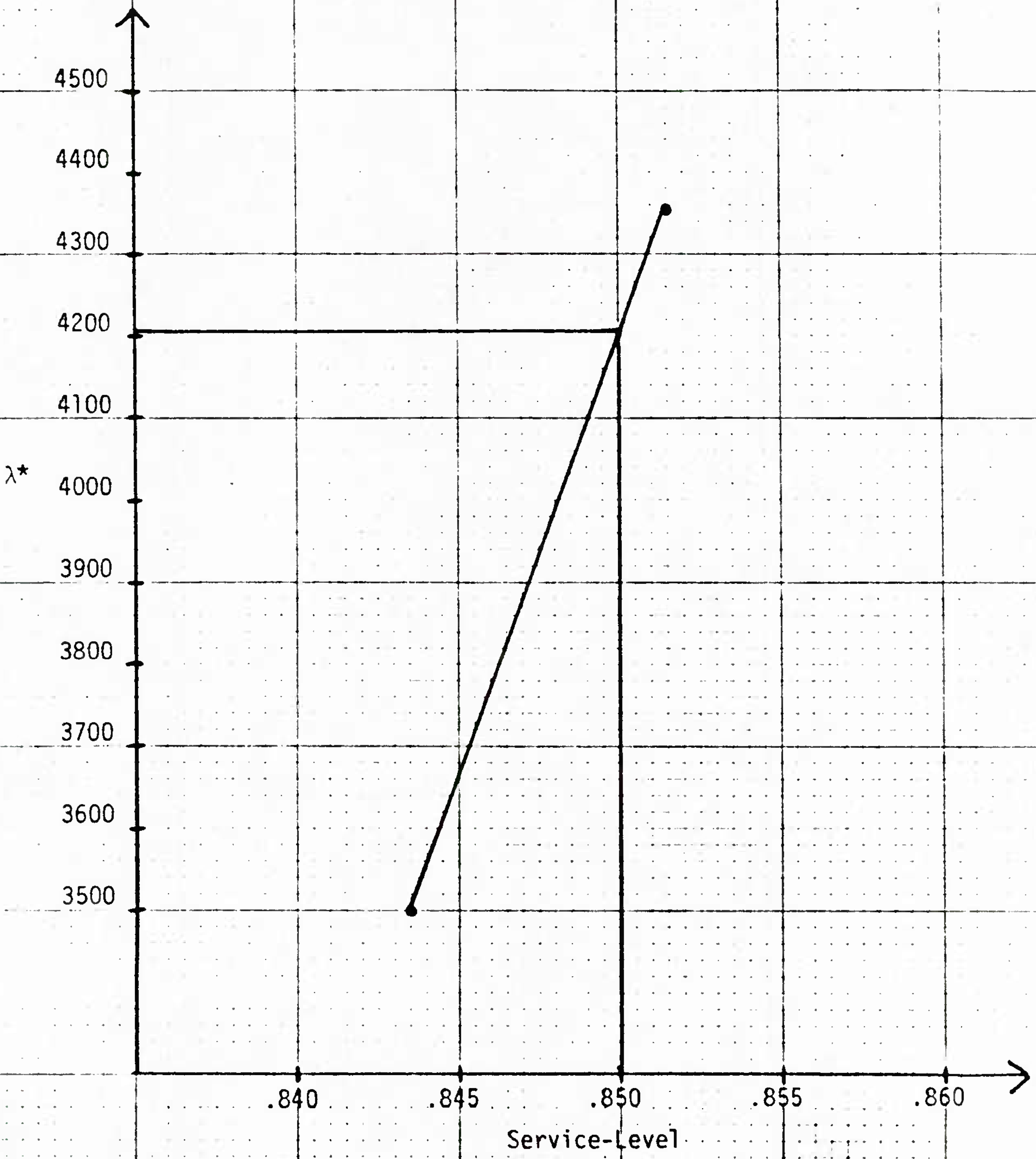


Figure 43. 32-Item Central Nonrandom Uniform Sample of 512-Item Inventory System.

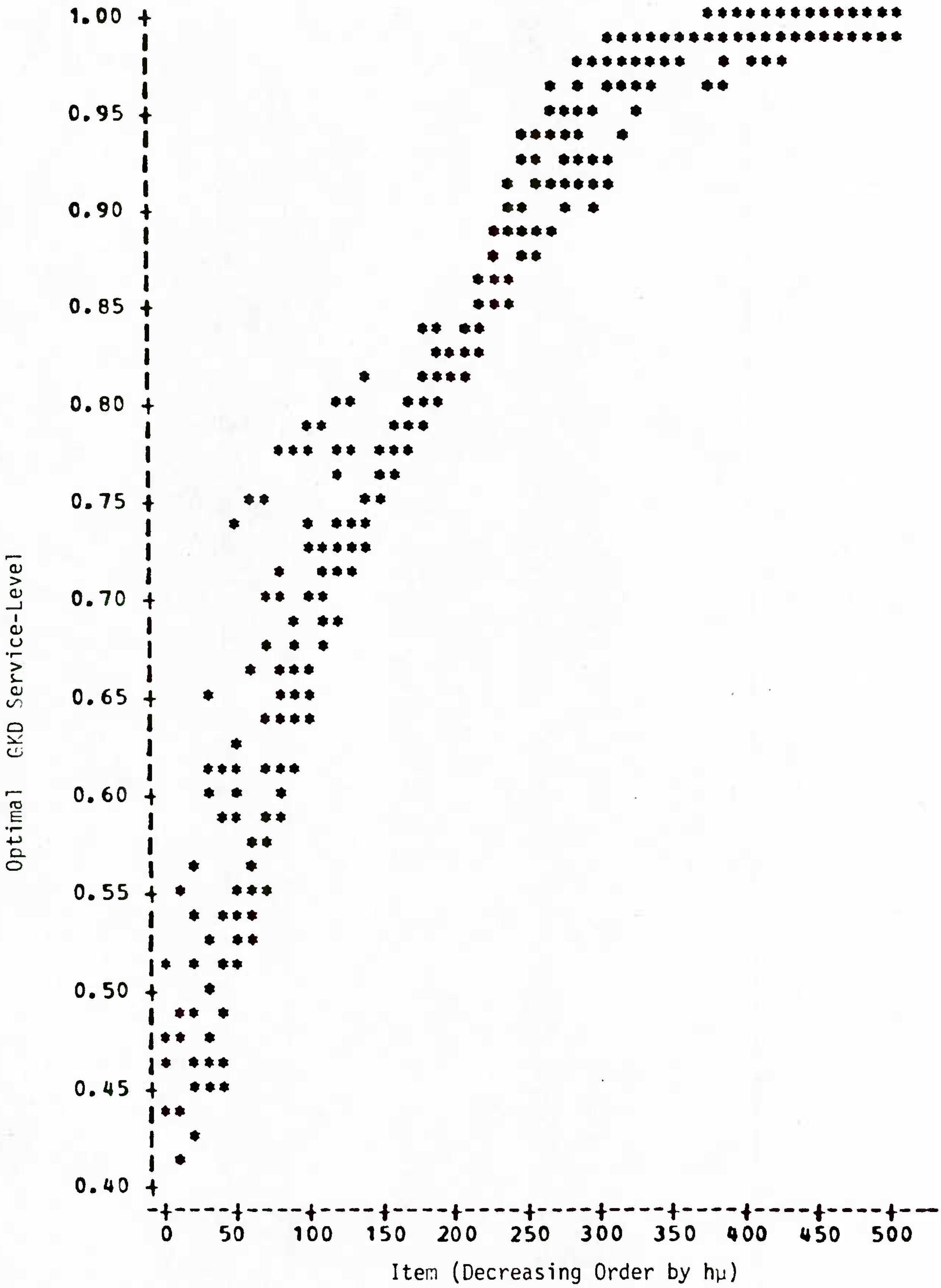


Figure 44. 512-Item Inventory System

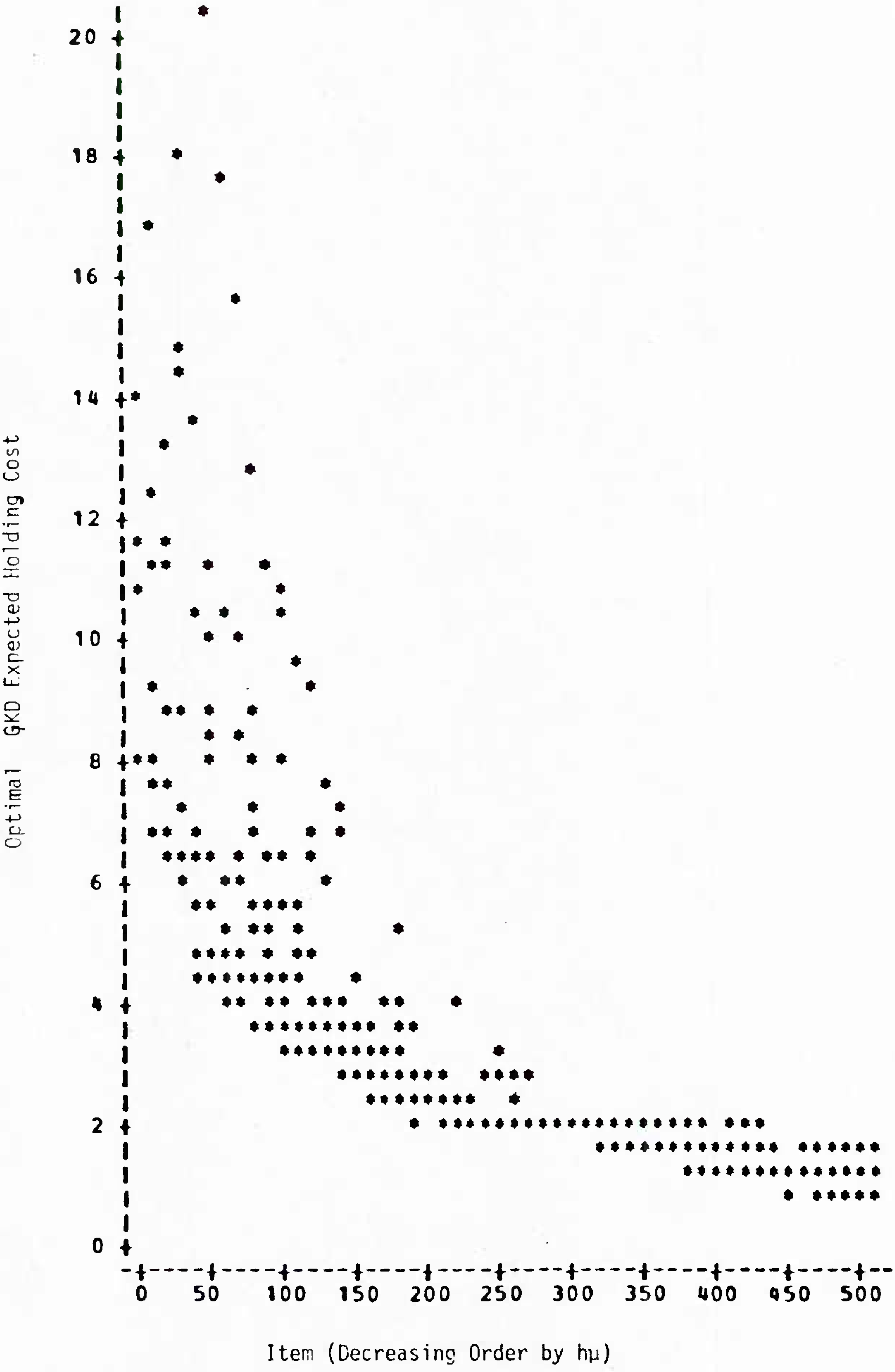


Figure 45. 512-Item Inventory System

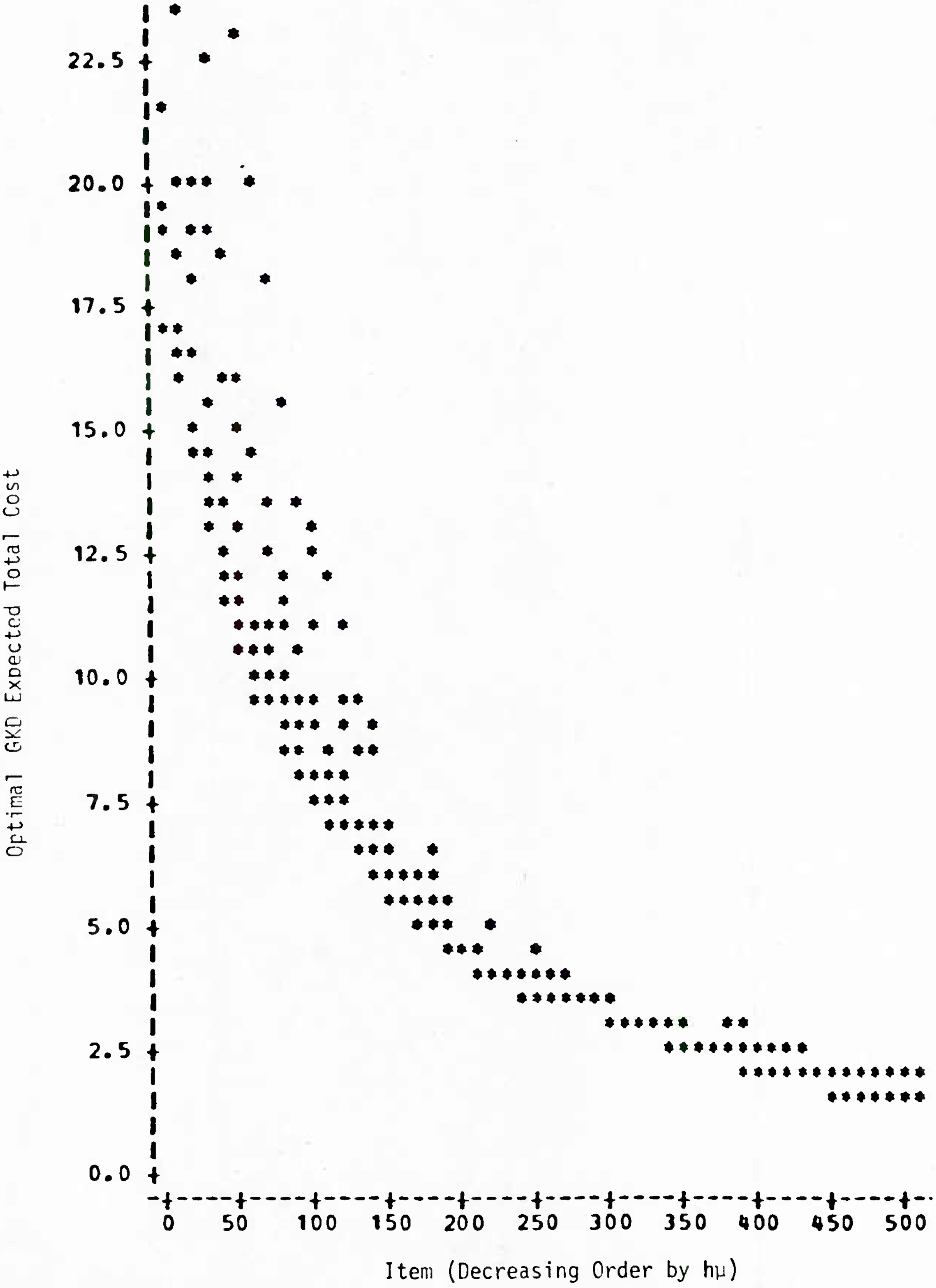


Figure 46. 512-Item Inventory System

APPENDIX TO CHAPTER V

This appendix contains a listing of the FORTRAN computer program for the programs LSMISS and MISSNLSI. The subroutines EXACT and POWAPP are the same for both of these programs, and the subroutine COMPAR is slightly different for the two programs.

```
C
C   PROGRAM LSMISS(NI) : LARGE - SCALE
C                        MULTI - ITEM (S,S)
C                        NORMAL DEMAND DISTRIBUTION APPROXIMATION
C                        LOWER BOUNDS ON BIG S
C
C    VERSION DATED 07/20/83
C
C   THE INPUT IS FROM CARDS
C
C    TITLE = USER - SUPPLIED TITLE (LE 36 ALPHA CHARACTERS)
C    NI = NUMBER OF ITEMS (NI LE 512)
C    K  = SET-UP ORDERING COST
C    L  = LEADTIME + 1
C    HO = UNIT HOLDING COST
C    MU = MEAN DEMAND
C    SD = DEMAND STANDARD DEVIATION
C    WT = CONSTRAINT WEIGHT
C
C    NSIG = NUMBER OF SIGNIFICANT DIGITS DESIRED FROM THE IMSL
C           ROUTINES ZBRENT AND ZFALSE
C    EPS = 1/(10 ** NSIG)
C    MAXFN = MAXIMUM NUMBER OF ITERATIONS ALLOWABLE BY USER FOR
C            THE IMSL ROUTINES ZBRENT AND ZFALSE
C    IER = (OUTPUT) ERROR PARAMETER FROM IMSL ROUTINES AND EXACT
C
C    THE FOLLOWING ARE INPUT TO OR COMPUTED BY GKD
C
C    LITS = LITTLE S
C    BIGS = BIG S
C    D = REORDER QUANTITY BIG S - LITTLE S
C    BSL = R * L * MU
C        = LOWER BOUND FOR BIG S (MAY BE INCREASED BY GKD)
C    AL = OVERALL SERVICE-LEVEL (0 < AL < 1)
C
C       THIS PROGRAM COMPUTES POLICES FOR
C       AL = LBAL, LBAL+INCRAL, LBAL+2*INCRAL, . . . ,UBAL
C
C    THE FOLLOWING ARE COMPUTED BY EHRHARDT'S POWER METHOD
C
C    PLITS = LITTLE S
C    PBIGS = BIG S
C
C    THE FOLLOWING ARE COMPUTED EXACTLY
C
C    HGKD = EXPECTED HOLDING COST FOR GKD POLICIES
C    TGKD = EXPECTED TOTAL COST FOR GKD POLICIES
C    AVSL  = WEIGHTED-AVERAGE SERVICE-LEVEL FOR GKD POLICIES
C
C    HPOW  = EXPECTED HOLDING COST FOR POWER POLICIES
```

```
C     TPOW   = EXPECTED TOTAL COST FOR POWER POLICIES
C     PAVSL  = WEIGHTED-AVERAGE SERVICE-LEVEL FOR POWER POLICIES
C
C    THE FOLLOWING PARAMETERS ARE USED IN THE STRATIFIED SAMPLING
C    SUBROUTINES STSAMP AND RNDSTS
C
C    SCHEME =   0   DO NOT USE ANY SAMPLING SCHEME
C               1   USE STRATIFED SAMPLING SCHEME
C               2   USE RANDOM STRATIFED SAMPLING SCHEME
C                                                                  -
C    SAMP     = NUMBER OF ITEMS IN THE SAMPLE                       |
C    NUMST    = NUMBER OF STRATA TO SAMPLE FROM                     |  ON
C    STA(I)   = LAST ITEM IN STRATA I                               |  INPUT
C    NUMB(I)  = NUMBER OF ITEMS FROM STATA I IN THE SAMPLE         _|
C
C    THE FOLLOWING VARIABLE IS NEEDED ON INPUT IF SCHEME = 2
C
C    SEED = SEED FOR PSEUDO-RANDOM NUMBER GENERATOR GGUD (IMSL)
C           REQUIRE SEED IN (1,2147483674)
C
C    ITEM(1), . . ., ITEM(SAMP) ARE THE ITEMS IN THE SAMPLE    ] OUTPUT
C
C    ENTIRE =   0   OTHERWISE
C               1   EVALUATE EXPECTED COSTS,ETC., FOR ENTIRE INVENTORY
C                   SYSTEM (THIS IS FOR TESTING)
C
C ***********************************************************************
C
C    INPUT THE SYSTEM PARAMETERS AS FOLLOWS:
C
C     TITLE
C    LBAL     UBAL     INCRAL    NSIG     MAXPN     NI
C    WT(1)     D(1)     K(1)     L(1)     HO(1)     MU(1)     SD(1)     R(1)
C    WT(2)     D(2)     K(2)     L(2)     HO(2)     MU(2)     SD(2)     R(2)
C      .        .        .        .         .         .         .         .
C      .        .        .        .         .         .         .         .
C      .        .        .        .         .         .         .         .
C    WT(N)     D(N)     K(N)     L(N)     HO(N)     MU(N)     SD(N)     R(N)
C    SCHEME   ENTIRE
C    SAMP     NUMST                            -
C    STRA(1)  STRA(2)  . . .  STRA(NUMST)     |   16 TO A
C    NUMB(1)  NUMB(2)  . . .  NUMB(NUMST)    _|     LINE
C    SEED    ]  ONLY IF SCHEME = 2
C
C    THE INPUT FORMAT IS:
C
C     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C    X.XXXXX X.XXXXX X.XXXXX XXXXXX XXXXXX XXXXXX
C    XXX.XXXX XXX.XXXX    .     .     .     XXX.XXXX XXX.XXXX
C    XXX.XXXX XXX.XXXX    .     .     .     XXX.XXXX XXX.XXXX
```

```
C     .        .        .    .    .       .        .
C     .        .        .    .    .       .        .
C     .        .        .    .    .       .        .
C  XXX.XXXX XXX.XXXX    .    .    .    XXX.XXXX XXX.XXXX
C  XX XX                          -
C  XXXX XXXX                       |
C  XXXX XXXX   .   .   .   XXXX    |  NOT NEEDED IF
C  XXXX XXXX   .   .   .   XXXX    |  SCHEME = 0
C  XXXXXXXXXX                     _|
C
C  ALL DATA STARTS IN COLUMN 1 (EXCEPT TITLE STARTS IN COLUMN 2),
C  AND THERE IS A SINGLE SPACE BETWEEN EACH DATA ENTRY
C
C **********************************************************************
C
      REAL*8 LITS(512),BIGS(512),BSL(512),D(512)
      REAL*8 BSL1(512),D1(512),BSLB(512)
      REAL*8 K(512),L(512),HO(512),MU(512),SD(512),WT(512)
      REAL*8 K1(512),L1(512),HO1(512),MU1(512),SD1(512),WT1(512)
      REAL*8 AL,EPS,LBAL,UBAL,INCRAL,LS(20),LSI,SLI,SUM,SEED,R
C
      INTEGER NI,N,M,NSIG,MAXFN,TITLE(8),J,NUM,ITEM(512),ENTIRE
      INTEGER SCHEME,SAMP,NUMST,STRA(512),NUMB(512),ISEED
C
      READ(1,3)  (TITLE(J),J=1,8)
    3 FORMAT(8A4)
      READ(1,5)  LBAL,UBAL,INCRAL,NSIG,MAXFN,NI
    5 FORMAT(3(F7.5,1X),3(I6,1X))
      EPS = 10.**(-NSIG)
C
      DO 10 I = 1,NI
      READ(1,8)  WT(I),D(I),K(I),L(I),HO(I),MU(I),SD(I),R
      BSL(I) = D(I)+R*L(I)*MU(I)
    8 FORMAT(8(F8.4,1X))
   10 CONTINUE
C
      READ(1,11)  SCHEME,ENTIRE
   11 FORMAT(I2,1X,I2)
      IF (SCHEME.EQ.0)  GO TO 19
      READ(1,15)  SAMP,NUMST
C
C    NUMST, STRA AND NUMB MUST BE CHANGED AS FOLLOWS FOR INPUT TO
C    THE SUBROUTINES STSAMP AND RNDSTS---SEE THE DOCUMENTATION FOR
C    THESE SUBROUTINES
C
      NUMST = NUMST+1
      STRA(1)  = 0
      NUMB(1)  = 0
      READ(1,15)  (STRA(I),I=2,NUMST)
      READ(1,15)  (NUMB(I),I=2,NUMST)
   15 FORMAT(16(I4,1X))
C
      ISEED = 0
      IF (SCHEME.EQ.2) READ(1,16) ISEED
   16 FORMAT(I10)
      SEED = ISEED
C
C   MAKE THE WEIGHTS WT INTO EQUIVALENT CONVEX WEIGHTS WT
C
   19 SUM = 0.
      DO 20 I = 1,NI
   20 SUM = SUM+WT(I)
      SUM = 1.0D0/SUM
      DO 25 I = 1,NI
   25 WT(I) = SUM*WT(I)
C
      M = 2+(UBAL-LBAL)/INCRAL
      IF (SCHEME.EQ.0) GO TO 300
C
C   CREATE THE INVENTORY SUBSYSTEM VIA STRATIFIED SAMPLING AND PUT
C   THE SUBSYSTEM IN N, K1, L1, . . . ,BSL1
C
      IF (SCHEME.EQ.1) CALL STSAMP(NI,K,L,HO,MU,SD,D,WT,BSL,N,K1,L1,
     &                             HO1,MU1,SD1,D1,WT1,BSL1,SAMP,
     &                             NUMST,STRA,NUMB,ITEM)
C
      IF (SCHEME.EQ.2) CALL RNDSTS(NI,K,L,HO,MU,SD,D,WT,BSL,N,K1,L1,
     &                             HO1,MU1,SD1,D1,WT1,BSL1,SAMP,
     &                             NUMST,STRA,NUMB,ITEM,SEED)
C
      DO 200 II = 1,M
      AL = LBAL+(II-1.D0)*INCRAL
      SLI = AL
      NUM = N
      DO 30 III = 1,N
   30 BSLB(III) = BSL1(III)
C
C   COMPUTE LAMBDA-STAR (LSI) USING THE GKD ALGORITHM WITH THE
C   INVENTORY SUBSYSTEM
C
      CALL GKD(NUM,L1,HO1,MU1,SD1,WT1,D1,SLI,LITS,BIGS,BSLB,EPS,
     &         NSIG,MAXPN,LSI)
      LS(II) = LSI
C
      WRITE(3,40) SCHEME,(STRA(I),I=2,NUMST)
   40 FORMAT(' RESULTS FOR INVENTORY SUBSYSTEM'//' SCHEME  ',I2//
     &       ' STRATA ',15(8I5/)/)
      WRITE(3,41) (NUMB(I),I=2,NUMST)
   41 FORMAT(' NUMBER ',15(8I5/)/)
      WRITE(3,42) (ITEM(I),I=1,SAMP)
   42 FORMAT(' ITEM    ',15(8I5/)/)
      WRITE(3,43) ISEED
   43 FORMAT(' SEED = ',I20/)
C
C   COMPUTE THE (UNIFORM SERVICE-LEVEL) POWER APPROXIMATION POLICIES
C   AND COMPARE THEIR ASSOCIATED COSTS WITH THE COSTS ASSOCIATED WITH
C   THE GKD POLICIES
C
      CALL COMPAR(N,K1,L1,HO1,MU1,SD1,WT1,BIGS,LITS,TITLE,LSI,AL)
      IF (ENTIRE.EQ.0) GO TO 200
C
C   COMPUTE THE OPTIMAL POLICIES ASSOCIATED WITH THE MULTIPLIER
C   LAMBDA-STAR (LSI)
C
      CALL BIGSLS(NI,L,HO,MU,SD,D,WT,BSL,LSI,EPS,NSIG,MAXFN,BIGS,LITS)
C
      WRITE(3,45)
   45 FORMAT('1')
      WRITE(3,50) SCHEME,(STRA(I),I=2,NUMST)
   50 FORMAT(' RESULTS FOR INVENTORY SYSTEM'//' SCHEME  ',I2//
     &       ' STRATA ',15(8I5/)/)
      WRITE(3,51) (NUMB(I),I=2,NUMST)
   51 FORMAT(' NUMBER ',15(8I5/)/)
      WRITE(3,52) (ITEM(I),I=1,SAMP)
   52 FORMAT(' ITEM    ',15(8I5/)/)
      WRITE(3,53) ISEED
   53 FORMAT(' SEED = ',I20/)
C
C   COMPUTE THE (UNIFORM SERVICE-LEVEL) POWER APPROXIMATION POLICIES
C   AND COMPARE THEIR ASSOCIATED COSTS WITH THE COSTS ASSOCIATED WITH
C   THE BIGSLS POLICIES
C
      CALL COMPAR(NI,K,L,HO,MU,SD,WT,BIGS,LITS,TITLE,LSI,AL)
  200 CONTINUE
      STOP
C
  300 DO 350 II = 1,M
      AL = LBAL+(II-1.D0)*INCRAL
      SLI = AL
      NUM = NI
      DO 320 III = 1,NI
  320 BSLB(III) = BSL(III)
C
C   COMPUTE LAMBDA-STAR (LSI) USING THE GKD ALGORITHM WITH THE
C   ENTIRE INVENTORY SYSTEM
C
      CALL GKD(NUM,L,HC,MU,SD,WT,D,SLI,LITS,BIGS,BSLB,EPS,
     &           NSIG,MAXFN,LSI)
C
C
```

```
C   COMPUTE THE (UNIFORM SERVICE-LEVEL) POWER APPROXIMATION POLICIES
C   AND COMPARE THEIR ASSOCIATED COSTS WITH THE COSTS ASSOCIATED WITH
C   THE GKD POLICIES
C
      CALL COMPAR(NI,K,L,HO,MU,SD,WT,BIGS,LITS,TITLE,LSI,AL)
  350 CONTINUE
      STOP
      END
C
      SUBROUTINE STSAMP(NI,K,L,HO,MU,SD,D,WT,BSL,N,K1,L1,HO1,MU1,
     &                  SD1,D1,WT1,BSL1,SAMP,NUMST,STRA,NUMB,ITEM)
C
C   STSAMP : STRATIFIED SAMPLING
C
C   THIS SUBROUTINE TAKES A STRATIFIED SAMPLE FROM THE INVENTORY
C   SYSTEM ACCORDING TO THE FOLLOWING PARAMETERS:
C
C   SAMP       = NUMBER OF ITEMS IN THE SAMPLE
C   NUMST      = NUMBER OF STRATA + 1
C   STRA(1)    = 0
C   STRA(I+1)  = LAST ITEM IN STRATA I, I = 1, 2, . . . , NUMST-1
C   NUMB(1)    = 0
C   NUMB(I+1)  = NUMBER OF ITEMS IN THE SAMPLE FROM STRATA I,
C                I = 1, 2, . . . , NUMST-1
C
C   FOR A GIVEN STRATA WITH NN ITEMS FROM WHICH M ARE TO BE
C   SAMPLED, THE ITEMS NN/2M, 3NN/2M, 5NN/2M, . . . , (2M-1)NN/2M
C   ARE TAKEN, WHERE THESE FRACTIONS ARE ROUNDED-OFF TO THE NEAREST
C   INTEGER
C   THESE ITEMS ARE DENOTED
C
C   ITEM(1), ITEM(2), . . ., ITEM(M)
C
C   THE CONSTRAINT WEIGHTS (WT), THE UNIT HOLDING COSTS (HO),
C   AND THE REPLENISHMENT SET-UP COSTS (K)
C   ARE RECOMPUTED TC REFLECT THE IDEA OF REPLACING ITEMS NOT IN
C   THE SAMPLE BY THOSE ITEMS NEAR THEM IN THE SAMPLE (IE, THOSE
C   IN THE SAME STRATA)
C   IN PARTICULAR, THESE COSTS ARE MULTIPLIED BY THE TOTAL
C   VALUE OF THE STRATA (VALUE = H*MU) DIVIDED BY THE TOTAL VALUE
C   OF THE SAMPLE FRCM THAT STRATA
C   THE WEIGHT FOR EACH ITEM IN A GIVEN STRATA IS THE SAME, BEING
C   THE TOTAL WEIGHT OF THE STRATA DIVIDED BY THE NUMBER OF ITEMS
C   SAMPLED FROM THE STRATA
C
      REAL*8 K(512),L(512),HO(512),MU(512),SD(512),WT(512),
     &       D(512),BSL(512)
      REAL*8 K1(512),L1(512),HO1(512),MU1(512),SD1(512),
     &       WT1(512),D1(512),BSL1(512)
      REAL*8 WEIGHT,RATIO,XM,XNSTR,HALF,ZERO,HWT,HSAMWT
```

```
C
      INTEGER SAMP,NUMST,STRA(512),NUMB(512),ITEM(512),FIRST,LAST
      INTEGER NI,N,I,J,IJ,IJIJ,JJ,NUMST,M,STRAI,NSTR
C
      DATA HALF,ZERO/0.5D0,0.0D0/
C
      N = SAMP
      IJ = 0
      DO 60 I = 2,NUMST
      STRAI = STRA(I-1)
      M = NUMB(I)
      XM = M
C
C   IF M = 0, THEN NO ITEM IN THE SAMPLE IS TO BE TAKEN FROM THIS
C   STRATA
C
      IF (M.EQ.0) GO TO 60
C
      NSTR = STRA(I)-STRAI
      XNSTR = NSTR
      RATIO = XNSTR/XM
      HWT = ZERO
      WEIGHT = ZERO
      DO 20 J = 1,NSTR
      JJ = STRAI+J
      HWT = HWT+HO(JJ)*MU(JJ)
   20 WEIGHT = WEIGHT+WT(JJ)
      WEIGHT = WEIGHT/XM
      HSAMWT = ZERO
      FIRST = IJ+1
      DO 40 J = 1,M
      IJ = IJ+1
      IJIJ = STRAI+(J-HALF)*RATIO+HALF
      ITEM(IJ)  = IJIJ
      K1(IJ)  = K(IJIJ)
      L1(IJ)  = L(IJIJ)
      HO1(IJ)  = HO(IJIJ)
      MU1(IJ)  = MU(IJIJ)
      SD1(IJ)  = SD(IJIJ)
      D1(IJ)  = D(IJIJ)
      BSL1(IJ)  = BSL(IJIJ)
      HSAMWT = HSAMWT+HO1(IJ)*MU1(IJ)
      WT1(IJ)  = WEIGHT
   40 CONTINUE
      LAST = FIRST+M-1
      HWT = HWT/HSAMWT
      DO 50 J = FIRST,LAST
      HO1(J)  = HWT*HO1(J)
   50 K1(J)  = HWT*K1(J)
   60 CONTINUE
```

```
      RETURN
      END
C
      SUBROUTINE RNDSTS(NI,K,L,HO,MU,SD,D,WT,BSL,N,K1,L1,HO1,MU1,
     &                  SD1,D1,WT1,BSL1,SAMP,NUMST,STRA,NUMB,ITEM,
     &                  SEED)
C
C  RNDSTS : RANDOM STRATIFIED SAMPLING
C
C  THIS SUBROUTINE TAKES A RANDOM STRATIFIED SAMPLE FROM THE
C  INVENTORY SYSTEM ACCORDING TO THE FOLLOWING PARAMETERS:
C
C  SAMP        = NUMBER OF ITEMS IN THE SAMPLE
C  NUMST       = NUMBER OF STRATA + 1
C  STRA(1)     = 0
C  STRA(I+1)   = LAST ITEM IN STRATA I, I = 1, 2, . . . , NUMST-1
C  NUMB(1)     = 0
C  NUMB(I+1)   = NUMBER OF ITEMS IN THE SAMPLE FROM STRATA I,
C                I = 1, 2, . . . , NUMST-1
C  SEED = SEED FOR PSEUDO-RANDOM NUMBER GENERATOR GGUD (IMSL)
C         REQUIRE SEED IN (1.0D0,2147483674.0D0)
C
C  FOR A GIVEN STRATA WITH NN ITEMS FROM WHICH M ARE TO BE
C  SAMPLED, THE ITEMS ARE SAMPLED RANDOMLY ACCORDING TO A DISCRETE
C  UNIFORM DISTRIBUTION
C  THESE ITEMS ARE DENOTED
C
C  ITEM(1), ITEM(2), . . ., ITEM(M)
C
C  THE CONSTRAINT WEIGHTS (WT), THE UNIT HOLDING COSTS (HO),
C  AND THE REPLENISHMENT SET-UP COSTS (K)
C  ARE RECOMPUTED TO REFLECT THE IDEA OF REPLACING ITEMS NOT IN
C  THE SAMPLE BY THOSE ITEMS NEAR THEM IN THE SAMPLE (IE, THOSE
C  IN THE SAME STRATA)
C  IN PARTICULAR, THESE COSTS ARE MULTIPLIED BY THE TOTAL
C  VALUE OF THE STRATA (VALUE = H*MU) DIVIDED BY THE TOTAL VALUE
C  OF THE SAMPLE FROM THAT STRATA
C  THE WEIGHT FOR EACH ITEM IN A GIVEN STRATA IS THE SAME, BEING
C  THE TOTAL WEIGHT OF THE STRATA DIVIDED BY THE NUMBER OF ITEMS
C  SAMPLED FROM THE STRATA
C
      REAL*8 K(512),L(512),HO(512),MU(512),SD(512),WT(512),
     &       D(512),BSL(512)
      REAL*8 K1(512),L1(512),HO1(512),MU1(512),SD1(512),
     &       WT1(512),D1(512),BSL1(512)
      REAL*8 WEIGHT,RATIO,XM,XNSTR,ZERO,HWT,HSAMWT,SEED,DSEED
C
      INTEGER SAMP,NUMST,STRA(512),NUMB(512),ITEM(512),FIRST,LAST
      INTEGER NI,N,I,J,IJ,IJIJ,JJ,NUMST,M,STRAI,NSTR,INDEX,ONE,
     &        IR(1)
```

```
C
      DATA ZERO,ONE/0.0D0,1/
C
      DSEED = SEED
      N = SAMP
      IJ = 0
      DO 60 I = 2,NUMST
      STRAI = STRA(I-ONE)
      M = NUMB(I)
      XM = M
C
C   IF M = 0, THEN NO ITEM IN THE SAMPLE IS TO BE TAKEN FROM THIS
C   STRATA
C
      IF (M.EQ.0) GO TO 60
C
      NSTR = STRA(I)-STRAI
      XNSTR = NSTR
      RATIO = XNSTR/XM
      HWT = ZERO
      WEIGHT = ZERO
      DO 20 J = ONE,NSTR
      JJ = STRAI+J
      HWT = HWT+HO(JJ)*MU(JJ)
   20 WEIGHT = WEIGHT+WT(JJ)
      WEIGHT = WEIGHT/XM
      HSAMWT = ZERO
      FIRST = IJ+ONE
      DO 40 J = ONE,M
   25 CALL GGUD(DSEED,NSTR,ONE,IR)
      IJIJ = STRAI+IR(ONE)
      DO 30 INDEX = ONE,IJ
      IF (IJIJ.EQ.ITEM(INDEX)) GO TO 25
   30 CONTINUE
      IJ = IJ+ONE
      ITEM(IJ) = IJIJ
      K1(IJ) = K(IJIJ)
      L1(IJ) = L(IJIJ)
      HO1(IJ) = HO(IJIJ)
      MU1(IJ) = MU(IJIJ)
      SD1(IJ) = SD(IJIJ)
      D1(IJ) = D(IJIJ)
      BSL1(IJ) = BSL(IJIJ)
      HSAMWT = HSAMWT+HO1(IJ)*MU1(IJ)
      WT1(IJ) = WEIGHT
   40 CONTINUE
      LAST = FIRST+M-ONE
      HWT = HWT/HSAMWT
      DO 50 J = FIRST,LAST
      HO1(J) = HWT*HO1(J)
   50 K1(J) = HWT*K1(J)
   60 CONTINUE
      RETURN
      END
C
      SUBROUTINE BIGSLS(NI,L,HO,MU,SD,D,WT,BSL,LSI,EPS,NSIG,MAXFN,
     &                  BIGS,LITS)
C
C  THIS SUBROUTINE COMPUTES THE OPTIMAL (S,S) POLICIES ASSOCIATED
C  WITH THE KUHN-TUCKER CONSTRAINT MULTIPLIER LAMBDA-STAR (LSI)
C
C
      COMMON /FINDTS/AI,BI,EI,FI,GI,HI,LAM
C
      REAL*8 LITS(512),BIGS(512)
      REAL*8 L(512),HO(512),MU(512),SD(512),WT(512),
     &       D(512),BSL(512)
      REAL*8 EPS,LSI,LI,HOI,MUI,SDI,DI,MSI,WTI,TSI,LAM,
     &       RHI,QI,GAI,DEI,AI,BI,EI,FI,GI,HI,VAR1,VAR2,ETI,TUI,
     &       KSI,LPTUI,BSLI,X,TUI,LB,UB,ZERO,XONE,HALF,CONST
C
      INTEGER NI,NSIG,MAXFN,MAX,ONE,CNTR
C
      REAL*8 LAMTH
      EXTERNAL LAMTH
C
      ZERO = 0.
      ONE  = 1
      XONE = 1.
      HALF = .5
      CONST = 1.700436904
C
      DO 60 I = ONE,NI
      HOI = HO(I)
      MUI = MU(I)
      LI  = L(I)
      MSI = LI*MUI
      SDI = SD(I)
      WTI = WT(I)
      DI = D(I)
      RHI = MUI/(DI+((MUI+(SDI*SDI/MUI))*HALF))
      QI  = XONE-RHI
      GAI = CONST/(LI*SDI)
      DEI = DEXP(GAI*DI)
      AI = DI*HOI
      VAR1 = XONE+DEI
      BI = AI*(QI+VAR1)
      EI = AI*(DEI+QI*VAR1)
      FI = AI*QI*DEI
      VAR1 = DI*RHI*GAI
      VAR2 = QI*(DEI-XONE)
      GI = WTI*(VAR1+VAR2)
      HI = WTI*(VAR1*DEI+VAR2)
C
C  DETERMINE BSL BY THE LARGEST VALUE OF THETA, TU, AND HENCE THE
C  SMALLEST VALUE OF BIG S, SO THAT D(LAMBDA)/D(THETA) LE 0,
C  GUARANTEEING THAT LAMBDA(THETA) IS INVERTIBLE ON (0, TU)
C  IF THE USER-SUPPLIED BSL IS GREATER THAN THIS, IT BECOMES THE
C  BSL USED
C
      ETI = GI*(BI*HI*HI-GI*(EI*HI-FI*GI))-AI*HI*HI*HI
C
C  IF ETA(I) IS LESS THAN 0, LAMBDA(THETA) MAY NOT BE CONVEX, AND
C  THERE IS NO GUARANTEE THAT THE GKD ALGORITHM WILL CONVERGE TO
C  A LOCAL MINIMUM---THE USER SHOULD EXERCISE CARE
C
      IF (ETI.LT.ZERO) WRITE(3,30) I,ETI
   30 FORMAT(' ETA(',I6,') =',E20.6,'   LT 0')
C
      CNTR = -1
   35 CNTR = CNTR+ONE
      TUI = (DEXP(GAI*(MSI-CNTR)))/DEI
      KSI = (FI/HI)-(AI/(GI*TUI*TUI))
      VAR1 = GI+HI*TUI
      LPTUI = KSI-(ETI/(GI*HI*VAR1*VAR1))
      IF (LPTUI.GE.ZERO) GO TO 35
C
      BSLI = CNTR+DI
      X = BSL(I)
      BSL(I) = BSLI
      IF (X.GT.BSLI) BSL(I) = X
      TUI = (DEXP(GAI*(MSI-(BSL(I)-DI))))/DEI
C
      LAM = LSI
      MAX = MAXFN
      LB = ZERO
      UB = TUI
C
C  IF LAMTH(UB) IS GREATER THAN ZERO, THEN THE FUNCTION
C  LAMBDA(THETA) FOR THIS ITEM IS ALWAYS GREATER THAN LAMBDA-STAR,
C  AND SO BIGS FOR THIS ITEM IS SET AT ITS LOWER-BOUND BSL
C
      IF (LAMTH(UB).GT.ZERO) GO TO 50
      CALL ZBRENT(LAMTH,EPS,NSIG,LB,UB,MAX,IER)
      TSI = UB
      BIGS(I) = MSI-((DLOG(TSI))/GAI)
C
C   BECAUSE OF ROUND-OFF ERROR, IT MAY BE THAT BIG S LT BSL.
C   IF SO, SET BIG S = BSL.
C
```

```
      IF (BIGS(I).LT.BSL(I)) BIGS(I) = BSL(I)
      LITS(I) = BIGS(I)-DI
      GO TO 60
   50 BIGS(I) = BSL(I)
      LITS(I) = BIGS(I)-DI
   60 CONTINUE
      RETURN
      END
      REAL FUNCTION LAMTH*8 (THETA)
C
      COMMON /FINDTS/AI,BI,EI,FI,GI,HI,LAM
      REAL*8 AI,BI,EI,FI,GI,HI,LAM,THETA
C
      LAMTH = AI+THETA*(BI-LAM*GI+THETA*(EI-LAM*HI+THETA*FI))
      RETURN
      END
C
      SUBROUTINE COMPAR(NI,K,L,HO,MU,SD,WT,BIGS,LITS,TITLE,LSI,AL)
C
C   THIS SUBROUTINE DOES THE FOLLOWING:
C
C   (1) EVALUATES EXACTLY THE HOLDING AND REPLENISHMENT COST, AND
C       THE SERVICE-LEVEL FOR THE POLICIES (LITS,BIGS) UNDER THE
C       ASSUMPTION OF NEGATIVE-BINOMIAL DEMAND (IN EXACT)
C
C   (2) COMPUTES APPROXIMATELY OPTIMAL (S,S) POLICIES WITH UNIFORM
C       SERVICE-LEVELS (IN POWAPP)
C
C   (3) EVALUATES THE POLICIES DESCRIBED IN (2) EXACTLY AS IN (1)
C
C   (4) PRINTS THE RESULTS OF STEPS (1) AND (3)
C
      REAL*8 LITS(512),BIGS(512)
      REAL*8 K(512),L(512),HO(512),MU(512),SD(512),WT(512)
      REAL*8 AL,HGKD,HPOW,TGKD,TPOW,LSI,LITSI,BIGSI
      REAL*8 KI,LI,HOI,MUI,SDI,SLI,HCI,RCI,AVSL,PAVSL
C
      INTEGER NI,TITLE(8)
      AVSL  = 0.
      HGKD = 0.
      TGKD = 0.
      DO 70 I = 1,NI
      KI = K(I)
      LI = L(I)
      HOI = HO(I)
      MUI = MU(I)
      SDI = SD(I)
      BIGSI = BIGS(I)
      LITSI = LITS(I)
      CALL EXACT(KI,LI,HOI,MUI,SDI,LITSI,BIGSI,SLI,HCI,RCI,IER)
      IF (IER.EQ.1) GO TO 300
      AVSL = AVSL+SLI*WT(I)
      HGKD = HGKD+HCI
      TGKD = TGKD+HCI+BCI
   70 CONTINUE
C
      PAVSL = 0.
      HPOW  = 0.
      TPOW  = 0.
      DO 75 I = 1,NI
      KI = K(I)
      LI = L(I)
      HOI = HO(I)
      MUI = MU(I)
      SDI = SD(I)
      CALL POWAPP(KI,LI,HOI,MUI,SDI,AVSL,LITSI,BIGSI)
      CALL EXACT(KI,LI,HOI,MUI,SDI,LITSI,BIGSI,SLI,HCI,BCI,IER)
      IF (IER.EQ.1) GO TO 300
      PAVSL = PAVSL+SLI*WT(I)
      HPOW  = HPOW+HCI
      TPOW  = TPOW+HCI+BCI
   75 CONTINUE
C
C  PRINT SUMMARY
C
      WRITE(3,98) (TITLE(J),J=1,8),LSI
   98 FORMAT(/T2,8A4,'        LAMBDA STAR = ',F12.4,//)
      WRITE(3,100) AL,AVSL,PAVSL,HGKD,HPOW,TGKD,TPOW
  100 FORMAT(T3,'SPECIFIED ALPHA      =',T27,F11.4,//,
     &       T3,'ACTUAL ALPHA (GKD) =',T27,F11.4,//,
     &       T3,'ACTUAL ALPHA (POW)  =',T27,F11.4,//,
     &       T3,'HOLDING COST (GKD) =',T27,F11.4,//,
     &       T3,'HOLDING COST (POW)  =',T27,F11.4,//,
     &       T3,'TOTAL COST (GKD)   =',T27,F11.4,//,
     &       T3,'TOTAL COST (POW)    =',T27,F11.4)
      RETURN
C
  300 WRITE(3,310) I
  310 FORMAT(' BIG S OR D > 2000 FOR ITEM',I6,/
     &       ' EXACT CANNOT EVALUATE THE (S,S) POLICY',//)
C
      RETURN
      END
C
      SUBROUTINE EXACT(K,L,H,MU,SD,LS,BS,SL,HC,RC,IER)
C
C   THIS SUBROUTINE COMPUTES THE SERVICE-LEVEL, HOLDING COST AND
C   REPLENISHMENT COST EXACTLY FOR THE POLICY (LS,BS) UNDER THE
C   ASSUMPTION OF NEGATIVE BINOMIAL DEMAND.
C
```

```
C    K = SET-UP COST
C    L = LEADTIME + 1
C    H = UNIT HOLDING COST
C    MU = MEAN DEMAND
C    SD = DEMAND STANDARD DEVIATION
C    LS = LITTLE S
C    BS = BIG S
C    SL = STEADY-STATE SERVICE-LEVEL (FREQUENCY OF PERIODS THAT
C         NO BACKORDER IS PLACED)
C    HC = EXPECTED HOLDING COST
C    RC = EXPECTED REPLENISHMENT COST
C
      REAL*8 K,L,H,MU,SD,LS,BS,SL,HC,RC,P,Q,R,RS,PROB(2000),HALF,
     &       NB(2000),M(2000),MI,RHO,VAR,SUM,PROBI,MD,ZERO,XONE,XTWO
      INTEGER S,D,UB1,UB2,ONE,TWO,IER
      DATA ONE,TWO,ZERO,XONE,XTWO,HALF/1,2,0.D0,1.D0,2.D0,.5D0/
C
C    S AND D ARE ROUNDED TO THE NEAREST INTEGER. WE USE THE REORDER
C    QUANTITY D = BIG S - LITTLE S - 1 BECAUSE WE USE THE VERSION OF
C    AN (S,S) POLICY THAT REQUIRES THAT AN ORDER BE PLACED WHEN THE
C    INVENTORY POSITION DROPS STRICTLY BELOW LITTLE S.
C
C    PROB(I)  = PROB(NB RV = I-1)
C    NB(I)    = PROB(NB + . . . + NB RV = I-1)
C                   |_____________|
C                          L
C    M(I)     = PROB(RENEWAL FNC OF NB RV = I-1)
C
C    IER = 0  EXACT WAS ABLE TO EVALUATE (S,S) POLICY
C          1  OTHERWISE
C
      IER = 0
      S = BS+HALF
      D = (BS-LS-ONE)+HALF
      Q = MU/(SD*SD)
      P = XONE-Q
      R = MU*Q/P
      RS = L*R
      VAR = Q**R
      PROB(ONE)  = VAR
      NB(ONE)    = Q**RS
      RHO        = XONE/(XONE-VAR)
      M(ONE)     = VAR*RHO
      UB1 = S+ONE
      IF (D.GT.S) UB1 = D+ONE
      IF (UB1.GT.2000) GO TO 100
      DO 20 I = TWO,UB1
      VAR = XONE/(I-XONE)
      PROBI = PROB(I-ONE)*P*(R+I-XTWO)*VAR
      PROB(I) = PROBI
      NB(I) = NB(I-ONE)*P*(RS+I-XTWO)*VAR
      SUM = ZERO
      UB2 = I-ONE
      DO 10 J = ONE,UB2
   10 SUM = SUM+PROB(I-J+ONE)*M(J)
      M(I) = (PROBI+SUM)*RHO
   20 CONTINUE
      UB1 = D+ONE
      IF (UB1.GT.2000) GO TO 100
      MD = ZERO
      DO 30 I = ONE,UB1
   30 MD = MD+M(I)
      RHO = XONE/(XONE+MD)
C
C   COMPUTE EXPECTED REPLENISHMENT COST
C
      RC = K*RHO
C
C   COMPUTE EXPECTED HOLDING COST
C
      IF (D.GT.S) UB1 = S+ONE
      IF (UB1.GT.2000) GO TO 100
      HC = ZERO
      DO 40 I = ONE,UB1
      UB2 = S-I+TWO
      MI = M(I)
      DO 40 J = ONE,UB2
      HC = HC+(UB2-J)*NB(J)*MI
   40 CONTINUE
      SUM = ZERO
      UB2 = S+ONE
      DO 45 I = ONE,UB2
   45 SUM = SUM+(UB2-I)*NB(I)
      HC = H*(HC+SUM)*RHO
C
C   COMPUTE (STEADY-STATE) SERVICE-LEVEL
C
      SL = ZERO
      DO 60 I = ONE,UB1
      SUM = ZERO
      UB2 = S-I+TWO
      DO 50 J = ONE,UB2
   50 SUM = SUM+NB(J)
   60 SL = SL+SUM*M(I)
      SUM = ZERO
      UB1 = S+ONE
      IF (UB1.GT.2000) GO TO 100
      DO 70 I = ONE,UB1
   70 SUM = SUM+NB(I)
      SL = (SL+SUM)*RHO
      RETURN
C
  100 IER = 1
      RETURN
C
      END
C
C
      SUBROUTINE PCWAPP(K,L,H,MU,SD,ALPHA,LITS,BIGS)
C
C    THIS SUBROUTINE COMPUTES LITTLE S AND BIG S USING THE REVISED
C    POWER APPROXIMATION METHOD (MOSIER(1981), TECH REPORT #18).
C    THE SHORTAGE COST IS COMPUTED USING THE POWER APPROXIMATION
C    METHOD (EHRHARDT(1977, PP 18, 45), TECH REPORT #12).
C
C    K = SET-UP COST
C    L = LEADTIME + 1
C    H = UNIT HOLDING COST
C    MU = MEAN DEMAND
C    SD = MEAN STANDARD DEVIATION
C    ALPHA = (STEADY-STATE) SERVICE-LEVEL, THE FREQUENCY OF PERIODS
C            WITHOUT BACKLOGS
C    LITS = LITTLE S
C    BIGS = BIG S
C
      REAL*8 P,H,K,MU,D,L,VL,SDL,Z,V,SD,LITS,BIGS,ONE,
     &       C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,ALPHA
C
      DATA C1/1.3/
      DATA C2/.494/
      DATA C3/.506/
      DATA C4/.116/
      DATA C5/.973/
      DATA C6/.183/
      DATA C7/1.063/
      DATA C8/2.192/
      DATA C10/.0695/
      DATA ONE/1./
C
      V  = SD*SD
      VL = V*L
      SDL = DSQRT(VL)
C
      P = H*(ALPHA-C10)/(ONE-ALPHA)
      Z = ONE+(VL/(MU*MU))
      D = C1*(MU**C2)*((K/H)**C3)*(Z**C4)
      Z = DSQRT(D*H/(P*SDL))
      C9 = (C6/Z)+C7-C8*Z
      LITS = C5*L*MU+SDL*C9
      BIGS = LITS+D
      RETURN
      END
C
      SUBROUTINE GKD(NI,L,HO,MU,SD,W,DD,AL,LITS,BIGS,BSL,EP,NS,MX,LS)
C
C   GKD (GENERALIZED KNAPSACK DUALITY) ALGORITHM
C
C   NOTE : THIS SUBROUTINE CHANGES THE VALUES OF THE INPUT
C          PARAMETER BSL
C
C   WT = WEIGHTS (MUST BE CONVEX)
C   RH = RHO
C   GA = GAMMA
C   DE = DELTA
C   A = A
C   B = B
C   E = E
C   F = F
C   G = G
C   H = H
C   MS = MU-STAR = L * MU
C   ET = ETA
C   KS = KSI
C   TU = THETA-UPPER
C   LPTU = LAMBDA-PRIME( THETA-UPPER )
C   LL = LAMBDA-LOWER
C   MLL = MAX( LAMBDA-LOWER )
C   PMLL = P( MLL )
C   LU = LAMBDA-UPPER
C   MLU = MAX( LAMBDA-UPPER )
C   TS = THETA-STAR
C   LS = LAMBDA-STAR
C   BSL = LOWER BOUND ON BIG S
C   MSL(I) = SERVICE-LEVEL FOR ITEM I
C   CONST = 15 * PI / (16 * SQRT(3))
C   NI = NUMBER OF ITEMS IN THE INVENTORY SYSTEM
C   N (AT PROGRAM BEGINNING) = NI
C   N (AT PROGRAM TERMINATION) = NUMBER OF ITEMS WITH BIG S GREATER
C                                THAN BSL
C   PERM : PERM(1), . . . , PERM(N) ARE THE ITEMS WITH BIG S
C          GREATER THAN BSL. WHEN NI IS GREATER THAN N,
C          PERM(N+1), . . . , PERM(NI) ARE THE ITEMS WITH
C          BIG S = BSL
C
      COMMON /INPUT/ WT,ALPHA,EPS,NSIG,MAXFN,N
      COMMON /LAM/ A,B,E,F,G,H
      COMMON /PARAMS/ TU,TS,RH,DE,GA,D,XONE,ZERO,PERM,ONE
      COMMON /SUBR/ LAM,I
C
      REAL*8 LITS(512),BIGS(512),BSL(512),W(512),DD(512)
      REAL*8 L(512),HO(512),MU(512),SD(512),WT(512),ALPHA
      REAL*8 D(512),RH(512),GA(512),DE(512),A(512),B(512),E(512),F(512),
     & G(512),H(512),TU(512),MS(512),LL(512),LU(512),BSL(512),
     & TS(512),MLL,EMLL,MLU,LS,ZERO,HALF,EPS,EP,AL
      REAL*8 CONST,HOI,MUI,LI,SDI,WTI,ALI,BI,EI,WTAVSL,
     & DI,RHI,QI,GAI,DEI,AI,BI,EI,FI,GI,HI,VAR1,VAR2,TUI,TSI,LU,
     & LPTUI,MSI,LAM,BIGSI,THS,TEST,SUM,X,
     & LB,UB,KSI,ETI,BSLI,TLI,LLI,LUI,XONE
C
      INTEGER N,ONE,NI,CNTR,PERM(512),ILB,J,INDEX,M,II
      INTEGER NS,NSIG,MAXFN,MAX
C
      REAL*8 LAMBDA,P,LTML
      EXTERNAL P,LTML
C
      ZERO = 0.
      ONE  = 1
      XONE = 1.
      HALF = .5
      CONST = 1.700436904
C
      ALPHA = AL
      EPS   = EP
      NSIG  = NS
      MAXFN = MX
      N     = NI
      DO 3 I = ONE,NI
    3 WT(I) = W(I)
C
      WRITE(3,5)
    5 FORMAT('1')
C
C    INITIALIZE THE PERMUTATION MATRIX PERM
C
      DO 6 I = ONE,NI
    6 PERM(I) = I
C
C     INITIALLY SET LAMBDA-STAR = 0, SINCE THIS IS HOW TO INTERPRET
C     LAMBDA-STAR IF ALL POLICIES ARE SUCH THAT BIGS = BSL
C
      LS = ZERO
C
C    COMPUTE THE CONSTANTS
C
      DO 20 I = ONE,NI
      HOI = HO(I)
      MUI = MU(I)
      LI  = L(I)
      MSI = LI*MUI
      MS(I) = MSI
      SDI = SD(I)
      WTI = WT(I)
      DI = DD(I)
      D(I) = DI
      RHI = MUI/(DI+((MUI+(SDI*SDI/MUI))*HALF))
      RH(I) = RHI
      QI  = XONE-RHI
      GAI = CONST/(LI*SDI)
      GA(I) = GAI
      DEI = DEXP(GAI*DI)
      DE(I) = DEI
      AI = DI*MUI
      A(I) = AI
      VAR1 = XONE+DEI
      BI = AI*(CI+VAR1)
      B(I) = BI
      EI = AI*(DEI+CI*VAR1)
      E(I) = EI
      FI = AI*QI*DEI
      F(I) = FI
      VAR1 = DI*RHI*GAI
      VAR2 = QI*(DEI-XONE)
      GI = WTI*(VAR1+VAR2)
      G(I) = GI
      HI = WTI*(VAR1*DEI+VAR2)
      H(I) = HI
C
C  DETERMINE BSL BY THE LARGEST VALUE OF THETA, TU, AND HENCE THE
C  SMALLEST VALUE OF BIG S, SO THAT D(LAMBDA)/D(THETA) LE 0,
C  GUARANTEEING THAT LAMBDA(THETA) IS INVERTIBLE ON (0, TU)
C  IF THE USER-SUPPLIED BSL IS GREATER THAN THIS, IT BECOMES THE
C  BSL USED
C
      ETI = GI*(BI*HI*HI-GI*(EI*HI-FI*GI))-AI*HI*HI*HI
C
C  IF ETA(I) IS LESS THAN 0, LAMBDA(THETA) MAY NOT BE CONVEX, AND
C  THERE IS NO GUARANTEE THAT THE GKD ALGORITHM WILL CONVERGE TO
C  A LOCAL MINIMUM---THE USER SHOULD EXERCISE CARE
C
      IF (ETI.LT.ZERO) WRITE(3,10) I,ETI
   10 FORMAT(' ETA(',I6,') =',E20.6,'    LT 0')
C
      CNTR = -1
    7 CNTR = CNTR+ONE
      TUI = (DEXP(GAI*(MSI-CNTR)))/DEI
      KSI = (FI/HI)-(AI/(GI*TUI*TUI))
      VAR1 = GI+HI*TUI
      LPTUI = KSI-(ETI/(GI*HI*VAR1*VAR1))
      IF (LPTUI.GE.ZERO) GO TO 7
C
```

```
      BSLI = CNTR+DI
      X = BSL(I)
      BSL(I) = BSLI
      IF (X.GT.BSLI) BSL(I) = X
      TUI = (DEXP(GAI*(MSI-(BSL(I)-DI))))/DEI
      TU(I) = TUI
C
      LLI = LAMBDA(TUI,I)
      LL(I) = LLI
   20 CONTINUE
C
C   COMPUTE MLL = MAX (LAMBDA-LOWER) = MAX (LAMBDA(TU)), SO ALL
C   FUNCTIONS LAMBDA(THETA) ARE INVERTIBLE ON [MLL,INFINITY)
C
   22 MLL = ZERO
      DO 30 II = ONE,N
      I = PERM(II)
      LLI = LL(I)
      IF (MLL.LT.LLI) MLL = LLI
   30 CONTINUE
C
      PMLL = P(MLL)
C
C   P(LAMBDA) IS INCRESING ON [MLL,INFINITY) WITH P(INFINITY)
C   GREATER THAN 0. CHECK IF P(MLL) IS LESS THAN OR EQUAL TO
C   0, SO SOLVE P(LAMBDA) = 0 FOR LAMBDA = LS ON [MLL,INFINITY)
C
      IF (PMLL.LE.ZERO) GO TO 50
C
C   P(MLL) IS GREATER THAN 0. LOCATE THAT ITEM ILB = PERM(INDEX)
C   WITH THE LARGEST VALUE OF LL = LAMBDA LOWER, AND SET
C   BIG S = BSL, IE, SET TS(= THETA-STAR) = TU(= THETA-UPPER)
C
      INDEX = ONE
      ILB = PERM(INDEX)
      DO 42 I = ONE,N
      J = PERM(I)
      IF (LL(J).GT.LL(ILB)) INDEX = I
      ILB = PERM(INDEX)
   42 CONTINUE
C
C   CHANGE N TO N - 1, AND REMOVE ITEM ILB FROM THE FIRST N ENTRIES
C   OF PERM AND PUT IT IN PERM(N+1)
C
      N = N-ONE
C
C   IF N = 0, ALL ITEMS ARE SUCH THAT BIG S = BSL
C
      IF (N.EQ.0) GO TO 85
      DO 43 I = INDEX,N
```

```
  43  PERM(I) = PERM(I+1)
C
      PERM(N+1) = ILB
C
C    SET TS = TU FOR ITEM ILB
C
      TS(ILB) = TU(ILB)
C
C   DETERMINE THE SERVICE-LEVEL ALI FOR ITEM ILB, AND RECOMPUTE THE
C   AVERAGE SERVICE-LEVEL (WHICH IS AGAIN DENOTED ALPHA) FOR THE
C   OTHER ITEMS SO THAT THE OVERALL AVERAGE SERVICE-LEVEL IS STILL
C   THE ORIGINALLY SPECIFIED ALPHA
C
C   NOTE : WHEN BIG S IS SET TO BSL FOR THE ITEM ILB, THE ITEM
C          IS NO LONGER USED IN THE COMPUTATION OF THE SOLUTION LS
C          TO P(LAMBDA) = 0. RATHER, THIS COMPUTATION IS DONE ONLY
C          FOR THOSE SUMMANDS OF P THAT CORRESPOND TO ITEMS WITH
C          BIG S GREATER THAN BSL. HENCE WE NEED ONLY DETERMINE
C          THE AVERAGE SERVICE-LEVEL FOR THESE ITEMS SO THAT THE
C          OVERALL SERVICE-LEVEL IS THE SPECIFIED ALPHA
C
      DEI = DE(ILB)
      GAI = GA(ILB)
      TSI = TS(ILB)
      RHI = RH(ILB)
      QI = XONE-RHI
      DI = D(ILB)
      WTI = WT(ILB)
      ALI = QI*(XONE+DLOG((XONE+TSI)/(XONE+
     &      DEI*TSI))/(GAI*DI))+(RHI/(XONE+TSI))
      ALPHA = (ALPHA-WTI*ALI)/(XONE-WTI)
C
      IF (ALPHA.LT.XONE) GO TO 22
      WRITE(3,47)
  47  FORMAT(' THE NEW AVERAGE SERVICE-LEVEL IS NOT LESS THAN',//,
     &       ' ONE, SO THE PROBLEM IS INFEASIBLE',//)
      GO TO 82
C
  50  VAR1 = -PMLL
C
C    SINCE P(MLL) LE 0, SOLVE P(LAMBDA) = 0 FOR LAMBDA = LS
C    ON [MLL,INFINITY). FIRST CHECK IF P(MLL) = 0 WITHIN EPSILON;
C    IF NOT, DETERMINE MLU LESS THAN INFINITY SO THAT P(MLU) IS
C    GREATER THAN OR EQUAL TO 0. THEN SOLVE P(LAMBDA) = 0 FOR
C    LAMBDA = LS ON [MLL,MLU]
C
      LS = MLL
      IF (VAR1.LE.EPS) GO TO 81
C
      MLU = MLL
      DO 80 II = ONE, N
      I = PERM(II)
      TLI = HALF*TU(I)
      RHI = RH(I)
      QI = XONE-RHI
      GAI = GA(I)
      DI  = D(I)
      DEI = DE(I)
C
C   FIND LU SUFFICENTLY LARGE SO THAT THE I-TH COMPONENT OF THE
C   FUNCTION P IS NON-NEGATIVE (HERE TEST IS THE I-TH COMPONENT
C   OF THE FUNCTION P), THEN SET MLU = MAX( LU )
C
   60 TEST = ((XONE-RHI)*(XONE+DLOG((XONE+TLI)/(XONE+
     & DEI*TLI))/(GAI*DI))+(RHI/(XONE+TLI))-ALPHA)
      IF (TEST.GT.ZERO) GO TO 70
      TLI = HALF*TLI
      GO TO 60
   70 LUI = LAMBDA(TLI,I)
      LU(I) =LUI
      IF (LUI.GT.MLU) MLU = LUI
   80 CONTINUE
C
C   SOLVE P(LAMBDA) = 0 FOR LAMBDA = LS IN [MLL,MLU]
C
      MAX = MAXFN
      LB = MLL
      UB = MLU
      CALL ZBRENT (P,EPS,NSIG,LB,UB,MAX,IER)
C
      LS = UB
   81 CONTINUE
C
C   COMPUTE THE (S,S) POLICIES FOR THOSE ITEMS WITH BIG S GREATER
C   THAN BSL
C
   82 DO 84 II = ONE, N
      I = PERM(II)
      GAI = GA(I)
      TSI = TS(I)
      MSI = MS(I)
      DI = D(I)
      BIGS(I) = MSI-((DLOG(TSI))/GAI)
C
C   BECAUSE OF ROUND-OFF ERROR, IT MAY BE THAT BIG S LT BSL.
C   IF SO, SET BIG S = BSL.
C
      IF (BIGS(I).LT.BSL(I)) BIGS(I) = BSL(I)
   84 LITS(I) = BIGS(I) - DI
C
```

```
C   COMPUTE THE (S,S) PCLICIES FOR THOSE ITEMS WITH BIG S EQUAL
C   TO BSL
C
      IF (NI.EQ.M) GO TO 88
   85 M = M+ONE
      DO 86 II = M,NI
      I = PERM(II)
      BIGS(I) = BSL(I)
   86 LITS(I) = BIGS(I)-D(I)
C
   88 WTAVSL = ZERO
C
C   COMPUTE THE SERVICE-LEVEL AND SHORTAGE COST FOR ALL THE ITEMS
C   IN THE INVENTORY SYSTEM, AND WTAVSL = WEIGHTED AVERAGE
C   SERVICE-LEVEL (HCPEFULLY THIS EQUALS APPROX THE ORIGINAL ALPHA)
C
      DO 90 II = ONE,NI
      I = PERM(II)
      DBI = DE(I)
      GAI = GA(I)
      TSI = TS(I)
      RHI = RH(I)
      QI = XONE-RHI
      MSI = MS(I)
      DI = D(I)
      BIGSI = BIGS(I)
      THS = DEXP(-GAI*(BIGSI-MSI))
      ALI = QI*(XONE+DLOG((XONE+TSI)/(XONE+
     &         DBI*TSI))/(GAI*DI))+(RHI/(XONE+TSI))
      MSL(I) = ALI
      WTAVSL = WTAVSL+ALI*WT(I)
   90 CONTINUE
C
      RETURN
      END
C
C
      REAL FUNCTION LAMBDA*8 (THETA,I)
C
      COMMON /LAM/ A,B,E,F,G,H
C
      REAL*8 A(512),B(512),E(512),F(512),G(512),H(512)
      REAL*8 THETA
C
      INTEGER I
C
      LAMBDA = (A(I)+THETA*(B(I)+THETA*(E(I)+THETA*F(I))))/
     & (THETA*(G(I)+THETA*H(I)))
      RETURN
      END
```

```
C
C
      REAL FUNCTION P*8 (LAM)
C
      COMMON /INPUT/ WT,ALPHA,EPS,NSIG,MAXFN,N
      COMMON /PARAMS/ TU,TS,RH,DE,GA,D,XONE,ZERO,PERM,ONE
      COMMON /SUBR/ XLAM,I
C
      REAL*8 WT(512),ALPHA
      REAL*8 D(512),RH(512),GA(512),DE(512),TU(512),TS(512),ZERO,EPS
      REAL*8 LAM,TSI,TUI,RHI,QI,LTML,TEST,LB,UB,XLAM,XONE
C
      INTEGER ONE,I,II,MAX,N,PERM(512),MAXFN
C
      EXTERNAL LTML
C
      XLAM = LAM
      P = ZERO
C
C   TO EVALUATE P(LAM), EVALUATE THETA, THE SOLUTION IN [0,TU] TO
C   LTML(THETA) = LAMBDA(THETA) - LAM = 0
C   FIRST CHECK IF LTML(TU) = 0
C
      DO 20 II = ONE, N
      I = PERM(II)
      TUI = TU(I)
C
C   SINCE LTML(TUI) IS LESS THAN OR EQUAL TO 0, IF IN THE COMPUTER
C   LTML(TUI) IS GREATER THAN OR EQUAL TO 0, THIS IMPLIES THAT
C   INDEED LTML(TUI) = 0
C
      TEST = LTML(TUI)
      IF (TEST.LT.ZERO) GO TO 5
      TSI = TUI
      TS(I) = TSI
      GO TO 10
    5 LB = ZERO
      UB = TUI
      MAX = MAXFN
      CALL ZFALSE(LTML,EPS,NSIG,LB,UB,TSI,MAX,IER)
      TS(I) = TSI
   10 RHI = RH(I)
      QI = XONE-RHI
      P = P+WT(I)*(QI*(XONE+DLOG((XONE+TSI)/(XONE+
     & DE(I)*TSI))/(GA(I)*D(I)))+(RHI/(XONE+TSI))-ALPHA)
   20 CONTINUE
C
      RETURN
      END
C
```

```
C
      REAL FUNCTION LTML*8 (THETA)
C
      COMMON /LAM/ A,B,E,F,G,H
      COMMON /SUBR/ LAM,I
C
      REAL*8 A(512),B(512),E(512),F(512),G(512),H(512)
      REAL*8 THETA,LAM
C
      INTEGER I
C
      LTML = A(I)+
     &       THETA*(B(I)-LAM*G(I)+THETA*(E(I)-LAM*H(I)+THETA*F(I)))
      RETURN
      END
```

```
C
C   PROGRAM MISSNLSI : MULTI - ITEM (S,S)
C                      NORMAL DEMAND DISTRIBUTION APPROXIMATION
C                      LOWER BOUNDS ON BIG S
C                      SINGLE-ITEM POLICY OUTPUT
C
C   THE INPUT IS FROM CARDS
C
C    TITLE = USER - SUPPLIED TITLE (LE 32 ALPHA CHARACTERS)
C    NI = NUMBER OF ITEMS (NI LE 512)
C    K = SET-UP ORDERING COST
C    L = LEADTIME + 1
C    HO = UNIT HOLDING COST
C    MU = MEAN DEMAND
C    SD = DEMAND STANDARD DEVIATION
C    WT = CONSTRAINT WEIGHT
C
C    NSIG = NUMBER OF SIGNIFICANT DIGITS DESIRED FROM THE IMSL
C           ROUTINES ZBRENT AND ZFALSE
C    EPS = 1/(10 ** NSIG)
C    MAXFN = MAXIMUM NUMBER OF ITERATIONS ALLOWABLE BY USER FOR
C            THE IMSL ROUTINES ZBRENT AND ZFALSE
C    IER = (OUTPUT) ERROR PARAMETER FROM IMSL ROUTINES AND EXACT
C
C    THE FOLLOWING ARE INPUT TO OR COMPUTED BY GKD
C
C    LITS = LITTLE S
C    BIGS = BIG S
C    D = REORDER QUANTITY BIG S - LITTLE S
C    BSL = D + R*L*MU
C        = LOWER BOUND FOR BIG S (MAY BE INCREASED BY GKD)
C    LS = LAMBDA-STAR
C
C    OUT = 0 OTHERWISE
C          1 WRITE OUTPUT TO THE EXTERNAL FILE
C            FORTRAN UNIT NUMBER 10
C            WITH THE FOLLOWING DCB SPECIFICATIONS:
C
C            RECFM   = FB
C            LRECL   = 160
C            BLKSIZE = 1600
C
C    THE FOLLOWING ARE COMPUTED BY EHRHARDT'S POWER METHOD
C
C    PLITS = LITTLE S
C    PBIGS = BIG S
C
C    THE FOLLOWING ARE COMPUTED EXACTLY
C
C    HC(I) = EXPECTED HOLDING COST FOR GKD POLICY FOR ITEM I
```

```
C    RC(I)  = EXPECTED REPLENISHMENT COST FOR GKD POLICY FOR ITEM I
C    SL(I)  = SERVICE LEVEL FOR GKD POLICY FOR ITEM I
C    AVSL   = WEIGHTED-AVERAGE SERVICE LEVEL FOR GKD POLICIES
C
C    PHC(I)  = EXPECTED HOLDING COST FOR POWER POLICY FOR ITEM I
C    PRC(I)  = EXPECTED REPLENISHMENT COST FOR POWER POLICY FOR ITEM I
C    PSL(I)  = SERVICE LEVEL FOR POWER POLICY FOR ITEM I
C    PAVSL   = WEIGHTED-AVERAGE SERVICE LEVEL FOR POWER POLICIES
C
C  ******************************************************************
C
C   INPUT THE SYSTEM PARAMETERS AS FOLLOWS:
C
C    TITLE
C   LS    NSIG    MAXFN    NI    OUT
C   WT(1)    D(1)    K(1)    L(1)    HO(1)    MU(1)    SD(1)    B(1)
C   WT(2)    D(2)    K(2)    L(2)    HO(2)    MU(2)    SD(2)    B(2)
C     .        .       .       .       .        .        .        .
C     .        .       .       .       .        .        .        .
C     .        .       .       .       .        .        .        .
C   WT(N)    D(N)    K(N)    L(N)    HO(N)    MU(N)    SD(N)    B(N)
C
C   THE INPUT FORMAT IS:
C
C    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C   XXXXXXXX.XXXXXX XXXXXX XXXXXX XXXXXX X
C   XXX.XXXX XXX.XXXX    .    .    .    XXX.XXXX XXX.XXXX
C   XXX.XXXX XXX.XXXX    .    .    .    XXX.XXXX XXX.XXXX
C      .        .        .    .    .       .        .
C      .        .        .    .    .       .        .
C      .        .        .    .    .       .        .
C   XXX.XXXX XXX.XXXX    .    .    .    XXX.XXXX XXX.XXXX
C
C   ALL DATA STARTS IN COLUMN 1 (EXCEPT TITLE STARTS IN COLUMN 2),
C   AND THERE IS A SINGLE SPACE BETWEEN EACH DATA ENTRY
C
C  ******************************************************************
C
C
      REAL*8 LITS(512),BIGS(512),BSL(512),D(512),PLITS(512),PBIGS(512)
      REAL*8 K(512),L(512),HO(512),MU(512),SD(512),WT(512),B
      REAL*8 AL,HMISS,HPOW,TMISS,TPOW,EPS
      REAL*8 KI,LI,HOI,MUI,SDI,SLI,HCI,RCI,AVSL,PAVSL,SUM
      REAL*8 SL(512),HC(512),RC(512),PSL(512),PHC(512),PRC(512)
      REAL*8 LSI,LITSI,BIGSI,DI,TEST(512)
C
      INTEGER NI
      INTEGER NSIG,MAXFN,ITEM,TITLE(8),OUT
C
      READ(1,3) (TITLE(J),J=1,8)
```

```
    3 FORMAT(8A4)
      READ(1,5) LSI,NSIG,MAXFN,NI,OUT
    5 FORMAT(F15.6,1X,3(I6,1X),I1)
      EPS = 10.**(-NSIG)
C
      DO 10 I = 1,NI
      READ(1,8) WT(I),D(I),K(I),L(I),HO(I),MU(I),SD(I),R
      BSL(I) = D(I)+R*L(I)*MU(I)
    8 FORMAT(8(F8.4,1X))
   10 CONTINUE
C
C    MAKE THE WEIGHTS WT INTO EQUIVALENT CONVEX WEIGHTS WT
C
      SUM = 0.
      DO 15 I = 1,NI
   15 SUM = SUM+WT(I)
      SUM = 1.0D0/SUM
      DO 16 I = 1,NI
   16 WT(I) = SUM*WT(I)
C
C   COMPUTE THE OPTIMAL POLICIES ASSOCIATED WITH THE MULTIPLIER
C   LAMBDA-STAR (LSI)
C
      CALL BIGSLS(NI,L,HO,MU,SD,D,WT,BSL,LSI,EPS,NSIG,MAXFN,BIGS,LITS)
C
      WRITE(3,45)
   45 FORMAT('1')
C
C   COMPUTE THE (UNIFORM SERVICE-LEVEL) POWER APPROXIMATION POLICIES
C   AND COMPARE THEIR ASSOCIATED COSTS WITH THE COSTS ASSOCIATED WITH
C   THE BIGSLS POLICIES
C
      CALL COMPAR(NI,K,L,HO,MU,SD,WT,BIGS,LITS,TITLE,LSI,OUT)
      STOP
      END
C
      SUBROUTINE BIGSLS(NI,L,HO,MU,SD,D,WT,BSL,LSI,EPS,NSIG,MAXFN,
     &                  BIGS,LITS)
C
C   THIS SUBROUTINE COMPUTES THE OPTIMAL (S,S) POLICIES ASSOCIATED
C   WITH THE KUHN-TUCKER CONSTRAINT MULTIPLIER LAMBDA-STAR (LSI)
C
C
      COMMON /FINDTS/AI,BI,EI,FI,GI,HI,LAM
C
      REAL*8 LITS(512),BIGS(512)
      REAL*8 L(512),HO(512),MU(512),SD(512),WT(512),
     &       D(512),BSL(512)
      REAL*8 EPS,LSI,LI,HOI,MUI,SDI,DI,MSI,WTI,TSI,LAM,
     &       RHI,QI,GAI,DEI,AI,BI,EI,FI,GI,HI,VAR1,VAR2,ETI,TUI,
     &        KSI,LPTUI,BSLI,X,TUI,LB,UB,ZERO,XONE,HALF,CONST
      REAL*8 HC(512),RC(512),SL(512),PHC(512),PRC(512),PSL(512)
      REAL*8 PLITS(512),PBIGS(512)
C
      INTEGER NI,NSIG,MAXFN,MAX,ONE,CNTR
C
      REAL*8 LAMTH
      EXTERNAL LAMTH
C
      ZERO = 0.
      ONE  = 1
      XONE = 1.
      HALF = .5
      CONST = 1.700436904
C
      DO 60 I = ONE,NI
      HOI = HO(I)
      MUI = MU(I)
      LI  = L(I)
      MSI = LI*MUI
      SDI = SD(I)
      WTI = WT(I)
      DI = D(I)
      RHI = MUI/(DI+((MUI+(SDI*SDI/MUI))*HALF))
      QI  = XONE-RHI
      GAI = CONST/(LI*SDI)
      DEI = DEXP(GAI*DI)
      AI = DI*HOI
      VAR1 = XONE+DEI
      BI = AI*(QI+VAR1)
      EI = AI*(DEI+QI*VAR1)
      FI = AI*QI*DEI
      VAR1 = DI*RHI*GAI
      VAR2 = QI*(DEI-XONE)
      GI = WTI*(VAR1+VAR2)
      HI = WTI*(VAR1*DEI+VAR2)
C
C   DETERMINE BSL BY THE LARGEST VALUE OF THETA, TU, AND HENCE THE
C   SMALLEST VALUE OF BIG S, SO THAT D(LAMBDA)/D(THETA) LE 0,
C   GUARANTEEING THAT LAMBDA(THETA) IS INVERTIBLE ON (0, TU)
C   IF THE USER-SUPPLIED BSL IS GREATER THAN THIS, IT BECOMES THE
C   BSL USED
C
      ETI = GI*(BI*HI*HI-GI*(EI*HI-FI*GI))-AI*HI*HI*HI
C
C   IF ETA(I) IS LESS THAN 0, LAMBDA(THETA) MAY NOT BE CONVEX, AND
C   THERE IS NO GUARANTEE THAT THE GRD ALGORITHM WILL CONVERGE TO
C   A LOCAL MINIMUM---THE USER SHOULD EXERCISE CARE
C
      IF (ETI.LT.ZERO) WRITE(3,30) I,ETI
```

```
   30  FORMAT(' ETA(',I6,') =',E20.6,'    LT 0')
C
       CNTR = -1
   35  CNTR = CNTR+ONE
       TUI = (DEXP(GAI*(MSI-CNTR)))/DEI
       KSI = (FI/HI)-(AI/(GI*TUI*TUI))
       VAR1 = GI+HI*TUI
       LPTUI = KSI-(ETI/(GI*HI*VAR1*VAR1))
       IF (LPTUI.GE.ZERO) GO TO 35
C
       BSLI = CNTR+DI
       X = BSL(I)
       BSL(I) = BSLI
       IF (X.GT.BSLI) BSL(I) = X
       TUI = (DEXP(GAI*(MSI-(BSL(I)-DI))))/DEI
C
       LAM = LSI
       MAX = MAXFN
       LB = ZERO
       UB = TUI
C
C   IF LAMTH(UB) IS GREATER THAN ZERO, THEN THE FUNCTION
C   LAMBDA(THETA) FOR THIS ITEM IS ALWAYS GREATER THAN LAMBDA-STAR,
C   AND SO BIGS FOR THIS ITEM IS SET AT ITS LOWER-BOUND BSL
C
       IF (LAMTH(UB).GT.ZERO) GO TO 50
       CALL ZBRENT(LAMTH,EPS,NSIG,LB,UB,MAX,IER)
       TSI = UB
       BIGS(I) = MSI-((DLOG(TSI))/GAI)
C
C    BECAUSE OF ROUND-OFF ERROR, IT MAY BE THAT BIG S LT BSL.
C    IF SO, SET BIG S = BSL.
C
       IF (BIGS(I).LT.BSL(I)) BIGS(I) = BSL(I)
       LITS(I) = BIGS(I)-DI
       GO TO 60
    50 BIGS(I) = BSL(I)
       LITS(I) = BIGS(I)-DI
    60 CONTINUE
       RETURN
       END
       REAL FUNCTION LAMTH*8 (THETA)
C
       COMMON /FINDTS/AI,BI,EI,FI,GI,HI,LAM
       REAL*8 AI,BI,EI,FI,GI,HI,LAM,THETA
C
       LAMTH = AI+THETA*(BI-LAM*GI+THETA*(EI-LAM*HI+THETA*FI))
       RETURN
       END
C
```

```
      SUBROUTINE COMPAR(NI,K,L,HO,MU,SD,WT,BIGS,LITS,TITLE,LSI,OUT)
C
C  THIS SUBROUTINE DOES THE FOLLOWING:
C
C  (1) EVALUATES EXACTLY THE HOLDING AND REPLENISHMENT COST, AND
C      THE SERVICE-LEVEL FOR THE POLICIES (LITS,BIGS) UNDER THE
C      ASSUMPTION OF NEGATIVE-BINOMIAL DEMAND (IN EXACT)
C
C  (2) COMPUTES APPROXIMATELY OPTIMAL (S,S) POLICIES WITH UNIFORM
C      SERVICE-LEVELS (IN POWAPP)
C
C  (3) EVALUATES THE POLICIES DESCRIBED IN (2) EXACTLY AS IN (1)
C
C  (4) PRINTS THE RESULTS OF STEPS (1) AND (3)
C
      REAL*8 LITS(512),BIGS(512)
      REAL*8 K(512),L(512),HO(512),MU(512),SD(512),WT(512)
      REAL*8 HGKD,HPOW,TGKD,TPOW,LSI,LITSI,BIGSI
      REAL*8 KI,LI,HOI,MUI,SDI,SLI,HCI,RCI,AVSL,PAVSL
      REAL*8 HC(512),RC(512),SL(512),PLITS(512),PBIGS(512),
     &       PHC(512),PRC(512),PSL(512)
C
      INTEGER NI,TITLE(8),OUT
      AVSL  = 0.
      HGKD = 0.
      TGKD = 0.
      DO 70 I = 1,NI
      KI = K(I)
      LI = L(I)
      HOI = HO(I)
      MUI = MU(I)
      SDI = SD(I)
      BIGSI = BIGS(I)
      LITSI = LITS(I)
      CALL EXACT(KI,LI,HOI,MUI,SDI,LITSI,BIGSI,SLI,HCI,RCI,IER)
      IF (IER.EQ.1) GO TO 300
      AVSL = AVSL+SLI*WT(I)
      HGKD = HGKD+HCI
      TGKD = TGKD+HCI+RCI
      HC(I) = HCI
      RC(I) = RCI
      SL(I) = SLI
   70 CONTINUE
C
      PAVSL = 0.
      HPOW  = 0.
      TPOW  = 0.
      DO 75 I = 1,NI
      KI = K(I)
      LI = L(I)
      HOI = HO(I)
      MUI = MU(I)
      SDI = SD(I)
      CALL POWAPP(KI,LI,HOI,MUI,SDI,AVSL,LITSI,BIGSI)
      CALL EXACT(KI,LI,HOI,MUI,SDI,LITSI,BIGSI,SLI,HCI,RCI,IER)
      IF (IER.EQ.1) GO TO 300
      PAVSL = PAVSL+SLI*WT(I)
      HPOW  = HPOW+HCI
      TPOW  = TPOW+HCI+RCI
      PLITS(I) = LITSI
      PBIGS(I) = BIGSI
      PHC(I)   = HCI
      PRC(I)   = RCI
      PSL(I)   = SLI
   75 CONTINUE
C
C  PRINT SUMMARY
C
      WRITE(3,98) (TITLE(J),J=1,8),LSI
   98 FORMAT(T2,8A4,'          LAMBDA-STAR = ',F15.6//)
      WRITE(3,100) AVSL,PAVSL,HGKD,HPOW,TGKD,TPOW
  100 FORMAT(T3,'ACTUAL ALPHA (GKD) =',T27,F11.4,//,
     &       T3,'ACTUAL ALPHA (POW)  =',T27,F11.4,//,
     &       T3,'HOLDING COST (GKD) =',T27,F11.4,//,
     &       T3,'HOLDING COST (POW)  =',T27,F11.4,//,
     &       T3,'TOTAL COST (GKD)   =',T27,F11.4,//,
     &       T3,'TOTAL COST (POW)    =',T27,F11.4,////,
     &       T4,'I',T8,'K',T12,'L+1',T19,'H',T25,'MU',
     &       T32,'SD',T40,'GKD',T50,'POWER',T65,'HOLD',
     &       T85,'REP',T99,'SER LEV',T116,'N*WT',//,
     &       T39,'LS',T44,'BS',T50,'LS',T55,'BS',
     &       T62,'GKD',T70,'POWER',T80,'GKD',T88,'POWER',
     &       T98,'GKD',T105,'POWER',//)
      DO 130 I = 1,NI
      SUM = NI*WT(I)
      WRITE(3,120) I,K(I),L(I),HO(I),MU(I),SD(I),LITS(I),BIGS(I),
     &             PLITS(I),PBIGS(I),HC(I),PHC(I),RC(I),PRC(I),
     &             SL(I),PSL(I),SUM
      WRITE(10,120) I,K(I),L(I),HO(I),MU(I),SD(I),LITS(I),BIGS(I),
     &             PLITS(I),PBIGS(I),HC(I),PHC(I),RC(I),PRC(I),
     &             SL(I),PSL(I),SUM
  120 FORMAT(I4,F5.0,F4.0,F9.4,F4.0,F9.4,2(F5.0,F6.0),4F9.2,
     &       2F8.4,F12.4,//)
  130 CONTINUE
      RETURN
C
  300 WRITE(3,310) I
  310 FORMAT(' BIG S OR D > 2000 FOR ITEM',I6,/
     &       ' EXACT CANNOT EVALUATE THE (S,S) POLICY',//)
C
```

```
      RETURN
      END
C
      SUBROUTINE EXACT(K,L,H,MU,SD,LS,BS,SL,HC,RC,IER)
C
C    THIS SUBROUTINE COMPUTES THE SERVICE-LEVEL, HOLDING COST AND
C    REPLENISHMENT COST EXACTLY FOR THE POLICY (LS,BS) UNDER THE
C    ASSUMPTION OF NEGATIVE BINOMIAL DEMAND.
C
C    K = SET-UP COST
C    L = LEADTIME + 1
C    H = UNIT HOLDING COST
C    MU = MEAN DEMAND
C    SD = DEMAND STANDARD DEVIATION
C    LS = LITTLE S
C    BS = BIG S
C    SL = STEADY-STATE SERVICE-LEVEL (FREQUENCY OF PERIODS THAT
C         NO BACKORDER IS PLACED)
C    HC = EXPECTED HOLDING COST
C    RC = EXPECTED REPLENISHMENT COST
C
      REAL*8 K,L,H,MU,SD,LS,BS,SL,HC,RC,P,Q,R,RS,PROB(2000),HALF,
     &      NB(2000),M(2000),RI,RHO,VAR,SUM,PROBI,MD,ZERO,XONE,XTWO
      INTEGER S,D,UB1,UB2,ONE,TWO,IER
      DATA ONE,TWO,ZERO,XONE,XTWO,HALF/1,2,0.D0,1.D0,2.D0,.5D0/
C
C    S AND D ARE ROUNDED TO THE NEAREST INTEGER. WE USE THE REORDER
C    QUANTITY D = BIG S - LITTLE S - 1 BECAUSE WE USE THE VERSION OF
C    AN (S,S) POLICY THAT REQUIRES THAT AN ORDER BE PLACED WHEN THE
C    INVENTORY POSITION DROPS STRICTLY BELOW LITTLE S.
C
C    PROB(I)  = PROB(NB RV = I-1)
C    NB(I)    = PROB(NB + . . . + NB RV = I-1)
C                    |______________|
C                           L
C    M(I)     = PROB(RENEWAL FNC OF NB RV = I-1)
C
C    IER = 0  EXACT WAS ABLE TO EVALUATE (S,S) POLICY
C          1  OTHERWISE
C
      IER = 0
      S = BS+HALF
      D = (BS-LS-ONE)+HALF
      Q = MU/(SD*SD)
      P = XONE-Q
      R = MU*Q/P
      RS = L*R
      VAR = Q**R
      PROB(ONE) = VAR
      NB(ONE)   = Q**RS
      RHO       = XONE/(XONE-VAR)
      M(ONE)    = VAR*RHO
      UB1 = S+ONE
      IF (D.GT.S) UB1 = D+ONE
      IF (UB1.GT.2000) GO TO 100
      DO 20 I = TWO,UB1
      VAR = XONE/(I-XONE)
      PROBI = PROB(I-ONE)*P*(R+I-XTWO)*VAR
      PROB(I) = PROBI
      NB(I) = NB(I-ONE)*P*(RS+I-XTWO)*VAR
      SUM = ZERO
      UB2 = I-ONE
      DO 10 J = ONE,UB2
   10 SUM = SUM+PROB(I-J+ONE)*M(J)
      M(I) = (PROBI+SUM)*RHO
   20 CONTINUE
      UB1 = D+ONE
      IF (UB1.GT.2000) GO TO 100
      MD = ZERO
      DO 30 I = ONE,UB1
   30 MD = MD+M(I)
      RHO = XONE/(XONE+MD)
C
C   COMPUTE EXPECTED REPLENISHMENT COST
C
      RC = K*RHO
C
C   COMPUTE EXPECTED HOLDING COST
C
      IF (D.GT.S) UB1 = S+ONE
      IF (UB1.GT.2000) GO TO 100
      HC = ZERO
      DO 40 I = ONE,UB1
      UB2 = S-I+TWO
      MI = M(I)
      DO 40 J = ONE,UB2
      HC = HC+(UB2-J)*NB(J)*MI
   40 CONTINUE
      SUM = ZERO
      UB2 = S+ONE
      DO 45 I = ONE,UB2
   45 SUM = SUM+(UB2-I)*NB(I)
      HC = H*(HC+SUM)*RHO
C
C   COMPUTE (STEADY-STATE) SERVICE-LEVEL
C
      SL = ZERO
      DO 60 I = ONE,UB1
      SUM = ZERO
      UB2 = S-I+TWO
      DO 50 J = ONE,UB2
   50 SUM = SUM+NB(J)
   60 SL = SL+SUM*M(I)
      SUM = ZERO
      UB1 = S+ONE
      IF (UB1.GT.2000) GO TO 100
      DO 70 I = ONE,UB1
   70 SUM = SUM+NB(I)
      SL = (SL+SUM)*RHO
      RETURN
C
  100 IER = 1
      RETURN
C
      END
C
C
      SUBROUTINE POWAPP(K,L,H,MU,SD,ALPHA,LITS,BIGS)
C
C    THIS SUBROUTINE COMPUTES LITTLE S AND BIG S USING THE REVISED
C    POWER APPROXIMATION METHOD (MOSIER(1981), TECH REPORT #18).
C    THE SHORTAGE COST IS COMPUTED USING THE POWER APPROXIMATION
C    METHOD (EHRHARDT(1977, PP 18, 45), TECH REPORT #12).
C
C    K = SET-UP COST
C    L = LEADTIME + 1
C    H = UNIT HOLDING COST
C    MU = MEAN DEMAND
C    SD = MEAN STANDARD DEVIATION
C    ALPHA = (STEADY-STATE) SERVICE-LEVEL, THE FREQUENCY OF PERIODS
C            WITHOUT BACKLOGS
C    LITS = LITTLE S
C    BIGS = BIG S
C
      REAL*8 P,H,K,MU,D,L,VL,SDL,Z,V,SD,LITS,BIGS,ONE,
     &       C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,ALPHA
C
      DATA C1/1.3/
      DATA C2/.494/
      DATA C3/.506/
      DATA C4/.116/
      DATA C5/.973/
      DATA C6/.183/
      DATA C7/1.063/
      DATA C8/2.192/
      DATA C10/.0695/
      DATA ONE/1./
C
      V  = SD*SD
      VL = V*L
      SDL = DSQRT(VL)
C
      P = H*(ALPHA-C10)/(ONE-ALPHA)
      Z = ONE+(VL/(MU*MU))
      D = C1*(MU**C2)*((K/H)**C3)*(Z**C4)
      Z = DSQRT(D*H/(P*SDL))
      C9 =  (C6/Z)+C7-C8*Z
      LITS = C5*L*MU+SDL*C9
      BIGS = LITS+D
      RETURN
      END
```

REFERENCES

1. Beesack, P.R. "A Finite Horizon Dynamic Inventory Model with a Stockout Constraint," Management Science, 13, 1967, pp. 618-630.

2. Bell. C. "Improved Algorithms for Inventory and Replacement-Stocking Problems," SIAM Journal of Applied Mathematics, 18, 1970, pp. 558-566.

3. Bellmore, M., H.J. Greenberg and J.J. Jarvis. "Generalized Penalty-Function Concepts in Mathematical Optimization," Operation Research, 18, 1970, pp. 229-252.

4. Bitran, G.R. and A.C. Hax. "Diagnostic Analysis of Inventory Systems: An Optimization Approach," Naval Research Logistics Quarterly, 29, 1982, pp. 29-46.

5. Brooks, R. and A.M. Geoffrian. "Finding Everett's Lagrange Multipliers by Linear Programming," Operations Research, 14, 1966, pp. 1149-1153.

6. Brown, R.G. Statistical Forecasting for Inventory Control. New York: McGraw Hill, 1959.

7. Brown, R.G. "Estimating Aggregate Inventory Standards," Naval Research Logistics Quarterly, 10, 1963, pp. 55-71.

8. Brown, R.G. Decision Rules for Inventory Management. New York: Holt, Rinehart and Winston, 1967.

9. Brown, R.G. Materials Management Systems. New York: John Wiley, 1977.

10. Denardo, E.V. Dynamic Programming. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1982.

11. Ehrhardt, R. "The Power Approximation: Inventory Policies Based on Limited Demand Information," ONR and ARO Technical Report #7, School of Organization and Management, Yale University, 1976.

12. Ehrhardt R. "Operating Characteristics Approximations for the Analysis of (s,S) Inventory Systems," Technical Report #12, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 1977.

13. Ehrhardt, R. "The Power Approximation for Computing (s,S) Inventory Policies," Management Science, 25, 1979, pp. 777-786.

14. Ehrhardt, R. "Analytic Approximations for (s,S) Inventory Policy Operating Characteristics," Naval Research Logistics Quarterly, 28, 1981, pp. 255-266.

15. Ehrhardt, R. and G. Kastner. "An Empirical Comparison of Two Approximately Optimal (s,S) Inventory Policies," Technical Report #16, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 1980.

16. Estey, A.S. and R.L. Kaufman. "Multi-Item Inventory System Policies Using Statistical Estimates: Negative Binomial Demands (Variance/Mean =9), ONR and ARO Technical Report #3, School of Organization and Management, Yale University, 1975.

17. Evans, R.V. "Inventory Control of a Multi-Product System With a Limited Production Resource," Naval Research Logistics Quarterly 14, 1967, pp. 173-184.

18. Everett, H. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," Operations Research, 11, 1963, pp. 399-413.

19. Federgruen, A. and P. Zipkin. "An Efficient Algorithm for Computing Optimal (s,S) Policies," Research Working Paper No. 458A, Graduate School of Business, Columbia University, New York, 1981.

20. Fox, B.L. "Discrete Optimization via Marginal Analysis," Management Science, 13, 1966, pp. 210-215.

21. Fox, B.L. and D.M. Landi. "Searching for the Multiplier in One-Constraint Optimization Problems," Operations Research, 1970, pp. 253-262.

22. Freeland, J.R. and E.L. Porteus. "Evaluating the Effectiveness of a New Method for Computing Approximately Optimal (s,S) Inventory Policies," Operations Research, 28, 1980, pp. 353-364.

23. Fromovitz, S. "A Class of One-Period Inventory Models," Operations Research, 13, 1965, pp. 779-799.

24. Gerson, G. and R.G. Brown. "Decision Rules for Equal Shortage Probabilities," Naval Research Logistics Quarterly, 17, 1970, pp. 351-358.

25. Greenberg, H. "Stock Level Distributions for (s,S) Inventory Problems," Naval Research Logistics Quarterly, 11, 1964, pp. 343-350.

26. Hadley, G. and T.M. Whitin. Analysis of Inventory Systems. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1963.

27. Hoffman, K. Analysis in Euclidian Space. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

28. Iglehart, D.L. "Optimality of (S,s) Policies in the Infinite Horizon Dynamic Inventory Problem," Management Science, 9, 1963a, pp. 259-267.

29. Iglehart, D.L. "Dynamic Programming and Stationary Analysis of Inventory Problems," Chapter 1 in Multistage Inventory Models and Techniques, H. Scarf, D. Gilford, M. Shelly (eds.), Stanford, C.A.: Stanford University Press, 1963b.

30. Iglehart, D.L. and S.C. Jaquette. "Optimal Policies Under the Shortage Probability Criterion for an Inventory Model with Unknown Dependent Demands," Naval Research Logistics Quarterly, 16, 1969, pp. 485-502.

31. Johnson, E.L. "On (s,S) Policies," Management Science, 15, 1968, pp. 80-101.

32. Johnson, N. and S. Kotz. Continuous Univariate Distributions. Boston, M.A.: Houghton Mifflin Co., 1970.

33. Kaplan, A.J. "A Note on EOQ Under Fund Constraints," Naval Research Logistics Quarterly, 25, 1978, pp. 369-374.

34. Karlin, S. "Optimal Inventory Policy for the Arrow-Harris-Marschak Dynamic Model," Chapter 9 in Studies in the Mathematical Theory of Inventory and Production, K. Arrow, S. Karlin, H. Scarf (eds.), Stanford, C.A.: Stanford Univ. Press, 1958a.

35. Karlin, S. "Steady State Solutions," Chapter 14 in Studies in the Mathematical Theory of Inventory and Production, 1958b.

36. Karlin, S. "The Application of Renewal Theory to the Study of Inventory Policies," Chapter 15 in Studies in the Mathematical Theory of Inventory and Production, 1958c.

37. Kastner, G. and R. Ehrhardt. "An Empirical Comparison of Approximately Optimal (s,S) Inventory Policies," ORSA Technical Report #79-2, University of North Carolina at Chapel Hill, 1979.

38. Kaufman, R. "Computer Programs for (s,S) Policies Under Independent or Filtered Demands," ONR and ARO Technical Report #5, School of Organization and Management, Yale University, 1976.

39. Klincewicz, J.G. "Biased Variance Estimators for Statistical Inventory Policies," ONR and ARO Technical Report #8, School of Organization and Management, Yale University, 1976a.

40. Klincewicz, J.G. "The Power Approximation: Control of Multi-Item Inventory Systems with Constant Standard Deviation-To-Mean Ratio for Demand," ONR and ARO Technical Report #10, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 1976b.

41. Leneman, O.A.Z. and F.J. Beutler. "On a New Approach to the Analysis of Stationary Inventory Problems," Naval Research Logistics Quarterly, 16, 1969, pp. 1-15.

42. Luenberger, D.G. Introduction to Linear and Non-Linear Programming. Reading, M.A.: Addison-Wesley Publishing Co., 1973.

43. MacCormick, A. "Statistical Problems in Inventory Control," ONR and ARO Technical Report #2, School of Organization and Management, Yale University, 1974.

44. MacCormick, A. "Inventory Control with Statistical Demand Information and a Shortage Probability Criterion," ORSA Technical Report #77-10, University of North Carolina at Chapel Hill, 1977.

45. MacCormick, A., A.S. Estey, and R.L. Kaufman. "Inventory Control with Statistical Demand Information," ORSA Technical Report #77-8, University of North Carolina at Chapel Hill, 1977.

46. Mitchell, J.C. "Choosing Single-Item Service Objectives in a Multi-Item Base-Stock Inventory System," Technical Report #20, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 1982.

47. Morse, P.M. "Solutions of a Class of Discrete Time Inventory Problems," Operations Research, 7, 1959, pp. 67-78.

48. Mosier, C. "Revised (s,S) Power Approximations," Technical Report #18, School of Business Administration and Curriculum in Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, 1981.

49. Naddor, E. "Optimal and Heuristic Decisions in Single- and Multi-Item Inventory Systems," Management Science, 21, 1975, pp. 1234-1249.

50. Norman, J. and D. White. "A Method for Approximate Solutions to Stochastic Dynamic Programming Problems Using Expectations," Operations Research, 16, 1968, pp. 296-306.

51. Parker, L.L. "Economical Reorder Quantities and Reorder Points With Uncertain Demand," Naval Research Logistics Quarterly, 11, 1964, pp. 351-358.

52. Peterson, R. and E.A. Silver. Decision Systems for Inventory Management and Production Planning, New York: John Wiley and Sons, 1979.

53. Porteus, E. "An Adjustment to the Norman-White Approach to Approximating Dynamic Programs," Operations Research, 27, 1979, pp. 1203-1208

54. Presutti, V. and R. Trepp. "More Ado About Economic Order Quantities," Naval Research Logistics Quarterly, 17, 1970, pp. 243-252.

55. Roberts, D. "Approximations to Optimal Policies in a Dynamic Inventory Model," in Studies in Applied Probability and Management Science, K. Arrow, S. Karlin and H. Scarf (eds.), Stanford, C.A.: Stanford University Press, 1962.

56. Ross, S.M. Applied Probability Models With Optimization Applications, San Francisco, C.A.: Holden-Day, Inc., 1970.

57. Sahin, I. "On the Objective Function Behavior in (s,S) Inventory Models," Operations Research, 30, 1982, pp. 709-724.

58. Scarf, H. "The Optimality of (S,s) Policies in the Dynamic Inventory Problem," Chapter 13 in Mathematical Methods in the Social Sciences, K. Arrow, S. Karlin and P. Suppes (eds), Stanford, C.A.: Stanford University Press, 1960.

59. Schäl, M. "On the Optimality of (s,S) Policies in Dynamic Inventory Models With Finite Horizon," SIAM Journal of Applied Mathematics, 30, 1976, pp. 528-537.

60. Schneider, H. "Methods for Determining the Re-Order Point of an (s,S) Ordering Policy When a Service Level is Specified," Journal of the Operational Research Society, 29, 1978, pp. 1181-1193.

61. Schrady, D.A. and V.C. Choe. "Models for Multi-Item Continuous Review Inventory Policies Subject to Constraints," Naval Research Logistics Quarterly, 18, 1971, pp. 451-464.

62. Schroeder, R.G. "Managerial Inventory Formulations With Stockout Objectives and Fiscal Constraints," Naval Research Logistics Quarterly, 21, 1974, pp.375-388.

63. Shapiro, J.F. "Nonlinearly Constrained Programming Problems: Generalized Lagrangian Functions and Penalty Methods," in Mathematical Programming: Structures and Algorithms, New York: John Wiley, 1979.

64. Sivazlian, B. "Dimensional and Computational Analysis in Stationary (s,S) Inventory Problems with Gamma Distributed Demand," Management Science, 17, 1971, pp. B307-B311.

65. Smith, W. "Asymptotic Renewal Theorems," Proc. Royal Soc. (Edinburgh), A, 64, 1954, pp. 9-48.

66. Tijms, H.C. and H. Groenevelt. "Simple Approximations for the Reorder Point in Periodic and Continuous Review (s,S) Inventory Systems With Service Level Constraints," Research Report No. 92, Vrije Universitat, 1982.

67. Veinott, A.F. "On the Optimality of (s,S) Inventory Policies: New Conditions and a New Proof," SIAM Journal of Applied Mathematics, 14, 1966a, pp. 1067-1083.

68. Veinott, A.F. "The Status of Mathematical Inventory Theory," Management Science, 12, 1966b, pp. 745-777.

69. Veinott, A.F. and H.M. Wagner, "Computing Optimal (s,S) Inventory Policies," Management Science, 11, 1965, pp. 525-552.

70. Wagner, H.M. Principles of Operations Research, Second Edition. Englewood Cliffs, N.J.: Prentice-Hall, 1975, pp. 831-836.

71. Wagner, H., M. O'Hagan and B. Lundh. "An Empirical Study of Exactly and Approximately Optimal Inventory Policies," Management Science, 11, 1965, pp. 690-723.

72. Winters, P.R. "Constrained Inventory Rules for Production Smoothing," Management Science, 8, 1962, pp. 470-481.

73. Zabel, E. "A Note on the Optimality of (S,s) Policies in Inventory Theory," Management Science, 9, 1962, pp. 123-125.